

કમ્પ્યુટર-અધ્યયન

ધોરણ 10



પ્રતિજ્ઞાપત્ર

ભારત મારો દેશ છે.
બધાં ભારતીયો મારાં ભાઈબહેન છે.
હું મારા દેશને ચાહું છું અને તેના સમૃદ્ધ અને
વૈવિધ્યપૂર્ણ વારસાનો મને ગર્વ છે.
હું સદાય તેને લાયક બનવા પ્રયત્ન કરીશ.
હું મારાં માતાપિતા, શિક્ષકો અને વડીલો પ્રત્યે આદર રાખીશ
અને દરેક જણ સાથે સવ્યતાથી વર્તીશ.
હું મારા દેશ અને દેશબાંધવોને મારી નિષ્ઠા અર્પું છું.
તેમનાં કલ્યાણ અને સમૃદ્ધિમાં જ મારું સુખ રહ્યું છે.

રાજ્ય સરકારની વિનામૂલ્યે યોજના હેઠળનું પુસ્તક



ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ
'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર-382010

© ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, ગાંધીનગર

આ પાઠ્યપુસ્તકના સર્વ હક ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળને હસ્તક છે.
આ પાઠ્યપુસ્તકનો કોઈ પણ ભાગ કોઈ પણ રૂપમાં ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળના નિયામકની લેખિત પરવાનગી વગર પ્રકાશિત કરી શકાશે નહિ.

વિષય-સલાહકાર

પ્રો. આર. પી. સોની

લેખન-સંપાદન

ડૉ. હર્ષલ આરોલકર (કન્વીનર)

ડૉ. સંજય એમ. શાહ

ડૉ. પ્રીતિ એસ. સજ્જા

ડૉ. કુંતલ પી. પટેલ

અનુવાદ

શ્રી રોહિત દોશી

શ્રી ગિરીશ એસ. બ્રહ્મભટ્ટ

શ્રી સાકેત એ. દવે

સમીક્ષા

શ્રી બિમલ કે. રાવલ

શ્રી રજનીકાન્ત એ. પંડ્યા

શ્રી પંકજ આર. શુક્લ

શ્રી રમણીક એલ. ગીલાતર

શ્રી રાજશ્રી એન. પડિયા

શ્રી સેજલ ડી. ત્રિવેદી

ભાષાશુદ્ધિ

શ્રી અશોકકુમાર એમ. દવે

સંયોજન

શ્રી આશિષ એચ. બોરીસાગર

(વિષય-સંયોજક : ગણિત)

નિર્માણ-આયોજન

શ્રી હરેશ એસ. લીઆચીયા

(નાયબ નિયામક : શૈક્ષણિક)

મુદ્રણ-આયોજન

શ્રી હરેશ એસ. લીઆચીયા

(નાયબ નિયામક : ઉત્પાદન)

પ્રસ્તાવના

માધ્યમિક અને ઉચ્ચતર માધ્યમિક શિક્ષણ બોર્ડે ઓપન સોર્સ ઓપરેટિંગ સિસ્ટમ અને તેને સુસંગત વિવિધ મુદ્દાઓ માટેના કમ્પ્યુટર-અધ્યયનને લગતા ઓપન સોર્સ સૉફ્ટવેર આધારિત નવો અભ્યાસક્રમ તૈયાર કર્યો છે. આ અભ્યાસક્રમ ગુજરાત સરકાર દ્વારા મંજૂર કરવામાં આવ્યો છે.

ગુજરાત સરકાર દ્વારા મંજૂર થયેલા **ધોરણ 10ના કમ્પ્યુટર-અધ્યયન** વિષયના નવા અભ્યાસક્રમ અનુસાર તૈયાર કરવામાં આવેલું આ પાઠ્યપુસ્તક વિદ્યાર્થીઓ સમક્ષ મૂકતાં મંડળ આનંદ અનુભવે છે.

આ વિષયનું અંગ્રેજી માધ્યમનું પાઠ્યપુસ્તક પ્રસિદ્ધ કરતાં પહેલાં એની હસ્તપ્રતની આ સ્તરે શિક્ષણકાર્ય કરતા શિક્ષકો અને તજજ્ઞો દ્વારા સર્વાંગી સમીક્ષા કરાવવામાં આવી છે અને તેમનાં સૂચનો અનુસાર હસ્તપ્રતમાં યોગ્ય સુધારા-વધારા કર્યા પછી આ પાઠ્યપુસ્તક પ્રસિદ્ધ કરવામાં આવ્યું છે. અંગ્રેજી માધ્યમના પાઠ્યપુસ્તકનો આ ગુજરાતી અનુવાદ છે.

પ્રસ્તુત પાઠ્યપુસ્તકને વિષયવસ્તુલક્ષી, રસપ્રદ અને ક્ષતિરહિત બનાવવા માટે મંડળે પૂરતી કાળજી લીધી છે, તેમ છતાં શિક્ષણમાં રસ ધરાવનાર વ્યક્તિઓ પાસેથી પુસ્તકની ગુણવત્તા વધારે તેવાં સૂચનો આવકાર્ય છે.

ડૉ. ભરત પંડિત

નિયામક

તા.3-3-2015

ડૉ. નીતિન પેથાણી

કાર્યવાહક પ્રમુખ

ગાંધીનગર

પ્રથમ આવૃત્તિ : 2014; પુનઃમુદ્રણ : 2014

પ્રકાશક : ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, 'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર વતી
ડૉ. ભરત પંડિત, નિયામક

મુદ્રક :

મૂળભૂત ફરજો

ભારતના દરેક નાગરિકની ફરજ નીચે મુજબ રહેશે :*

- (ક) સંવિધાનને વફાદાર રહેવાની અને તેના આદર્શો અને સંસ્થાઓનો, રાષ્ટ્રધ્વજનો અને રાષ્ટ્રગીતનો આદર કરવાની;
- (ખ) આઝાદી માટેની આપણી રાષ્ટ્રીય લડતને પ્રેરણા આપનારા ઉમદા આદર્શોને હૃદયમાં પ્રતિષ્ઠિત કરવાની અને અનુસરવાની;
- (ગ) ભારતનાં સાર્વભૌમત્વ, એકતા અને અખંડિતતાનું સમર્થન કરવાની અને તેમનું રક્ષણ કરવાની;
- (ઘ) દેશનું રક્ષણ કરવાની અને રાષ્ટ્રીય સેવા બજાવવાની હાકલ થતાં, તેમ કરવાની;
- (ચ) ધાર્મિક, ભાષાકીય, પ્રાદેશિક અથવા સાંપ્રદાયિક ભેદોથી પર રહીને, ભારતના તમામ લોકોમાં સુમેળ અને સમાન બંધુત્વની ભાવનાની વૃદ્ધિ કરવાની, જીઓના ગૌરવને અપમાનિત કરે, તેવા વ્યવહારો ત્યજી દેવાની;
- (છ) આપણી સમન્વિત સંસ્કૃતિના સમૃદ્ધ વારસાનું મૂલ્ય સમજી તે જાળવી રાખવાની;
- (જ) જંગલો, તળાવો, નદીઓ અને વન્ય પશુપક્ષીઓ સહિત કુદરતી પર્યાવરણનું જતન કરવાની અને તેની સુધારણા કરવાની અને જીવો પ્રત્યે અનુકંપા રાખવાની;
- (ઝ) વૈજ્ઞાનિક માનસ, માનવતાવાદ અને જિજ્ઞાસા તથા સુધારણાની ભાવના કેળવવાની;
- (ટ) જાહેર મિલકતનું રક્ષણ કરવાની અને હિંસાનો ત્યાગ કરવાની;
- (ઠ) રાષ્ટ્ર પુરુષાર્થ અને સિદ્ધિનાં વધુ ને વધુ ઉન્નત સોપાનો ભણી સતત પ્રગતિ કરતું રહે એ માટે, વૈયક્તિક અને સામૂહિક પ્રવૃત્તિનાં તમામ ક્ષેત્રે શ્રેષ્ઠતા હાંસલ કરવાનો પ્રયત્ન કરવાની;
- (ડ) માતા-પિતાએ અથવા વાલીએ ૬ વર્ષથી ૧૪ વર્ષ સુધીની વયના પોતાના બાળક અથવા પાલ્યને શિક્ષણની તકો પૂરી પાડવાની.

અનુક્રમણિકા

1. HTML નો પરિચય	1
2. HTMLમાં Head અને Body વિભાગ	18
3. HTMLમાં છબીઓનું વ્યવસ્થાપન	37
4. HTMLમાં યાદી અને કોષ્ટકનો ઉપયોગ	55
5. કેસીનો પરિચય	81
6. કેસીમાં ડેટાનું ઓડિટિંગ અને ફોર્મેટિંગ	97
7. કેસીમાં વિધેય	127
8. કેસીમાં આલેખની રચના	153
9. સમસ્યા અને સમસ્યાનું નિરાકરણ	175
10. સી ભાષાનો પરિચય	192
11. સી ભાષામાં ડેટા પ્રકાર, પ્રક્રિયકો અને પદાર્થો	216
12. નિવેશ(Input) / નિર્ગમ (Output) પ્રક્રિયાઓનો ઉપયોગ	239
13. નિર્ણય માળખાં	255
14. લૂપ નિયંત્રણ માળખાં	270
15. એરે	284
16. વિધેય	296
• પરિશિષ્ટ - I	309
• પરિશિષ્ટ - II	310
• પરિશિષ્ટ - III	311
• પરિશિષ્ટ - IV	312

પ્રિય શિક્ષકમિત્રો,

કમ્પ્યુટર-સાક્ષરતાનો દ્રુતગતિએ ફેલાવો કરવાના ધ્યેય સાથે ગુજરાત સરકારે ICT@School કાર્યક્રમ હેઠળ 6000થી વધુ શાળાઓને અદ્યતન કમ્પ્યુટર-સામગ્રી પૂરી પાડી છે. નવી નીતિની પહેલ તરીકે તમામ શાળાઓને ઉબન્ટુ (લિનક્સનું એક સ્વરૂપ) ઓપરેટિંગ સિસ્ટમ અને અન્ય ઓપન સોર્સ સોફ્ટવેર પેકેજ આપવામાં આવ્યાં છે, જેથી શાળાઓ લાઈસન્સ જેવી કોઈ પણ સમસ્યાઓની ચિંતા કર્યા વિના સોફ્ટવેરનો ઉપયોગ કરી શકે કે તેને બદલી શકે. અગાઉનાં પાઠ્યપુસ્તકો મોટા ભાગે માલિકીહક ધરાવતાં સોફ્ટવેર પર આધારિત હોઈ નવા અભ્યાસક્રમ આધારિત પાઠ્યપુસ્તકો ફરી તૈયાર કરવાની આવશ્યકતા જણાઈ. આ ઉપરાંત ધોરણ 8ને પ્રાથમિક વિભાગમાં સ્થાનાંતરિત કરવામાં આવ્યું હોવાથી પણ આ અનિવાર્ય હતું. આથી, કમ્પ્યુટર-અધ્યયનના વિવિધ વિષયો માટે ઓપન સોર્સ ઓપરેટિંગ સિસ્ટમ અને તેને સુસંગત ઓપન સોર્સ સોફ્ટવેર ટૂલ્સ આધારિત નવું વિષયવસ્તુ ધોરણ 9થી12 માટે ક્રમાનુસાર પૂરું પાડવામાં આવ્યું છે.

ધોરણ 10નું આ પાઠ્યપુસ્તક ‘કમ્પ્યુટર-અધ્યયન’ વિષયની શ્રેણીમાં બીજું પુસ્તક છે. ધોરણ-9માં વિદ્યાર્થીઓ ઓપન સોર્સ ઓપરેટિંગ સિસ્ટમ ઉબન્ટુ, ઓપન ઓફિસ શબ્દ પ્રક્રિયક રાઈટર, પ્રસ્તુતિ માટેનું ટૂલ ઈમ્પ્રેસ, ઈન્ટરનેટના ઉપયોગને લગતી માહિતીની શોધ, ઈ-મેઈલ, ફાઈલ ડાઉનલોડિંગ વગેરેના અભિગમો તથા સલામતીનાં પાસાં વિશે અભ્યાસ કરી ચૂક્યા છે. દસમા ધોરણના આ પાઠ્યપુસ્તકમાં તેમને શરૂઆતમાં વેબપેજ અને તેમાં માહિતીની રજૂઆત માટે HTML નામની મૂળભૂત માર્કઅપ ભાષાનો પરિચય કરાવવામાં આવ્યો છે. ઓપન ઓફિસમાં કેલ્સી નામના સ્પ્રેડશીટ ઘટકની પણ વિસ્તૃત ચર્ચા કરવામાં આવી છે. ત્યાર પછી તેઓ સમસ્યાના ઉકેલની પદ્ધતિઓ અને સી પ્રોગ્રામિંગ ભાષા વિશે અભ્યાસ કરશે. સી ભાષાની સરળ સમજૂતી માટે અનેક ઉદાહરણો તેનાં વિવરણ સહિત આપવામાં આવ્યાં છે. અમારું માનવું છે કે, પ્રોગ્રામિંગ ભાષાનો અભ્યાસ વિદ્યાર્થીની તર્કક્ષમતાનો વિકાસ કરવામાં અવશ્ય મદદરૂપ સાબિત થશે.

અમને આશા છે કે, આ પ્રયાસ ધોરણ 10ના વિદ્યાર્થીઓને ઉપયોગી નીવડશે અને આપ પણ ઓપન સોર્સ ઉબન્ટુ ઓપરેટિંગ સિસ્ટમ અને ટૂલ્સના ઉપયોગ દ્વારા પ્રાયોગિક કાર્ય અને શિક્ષણનો આનંદ મેળવશો.

પ્રિય વિદ્યાર્થીઓ,

કમ્પ્યુટરને લગતી વિવિધ વિભાવનાઓ, ઓપરેટિંગ સિસ્ટમ ઉબન્ટુ તથા અન્ય પ્રચલિત ઓપરેટિંગ સિસ્ટમ, શબ્દ-પ્રક્રિયક અને પ્રસ્તુતિ માટેના ઓપન ઓફિસનાં ઘટકો, ઈન્ટરનેટ તથા તેના ઉપયોગથી તમે સુપરિચિત છો ત્યારે હવે પછીનો સમય એનાથી આગળ વધીને HTML, સ્પ્રેડશીટ કેલ્સી અને સી પ્રોગ્રામિંગ ભાષા જેવા અદ્યતન મુદ્દાઓનો અભ્યાસ કરવાનો છે.

ધોરણ 10ના આ પાઠ્યપુસ્તકનાં પ્રકરણ 1 થી 4માં માર્કઅપ ભાષા HTMLના અગત્યના મુદ્દાઓનો સમાવેશ કરવામાં આવ્યો છે. આ પ્રકરણોમાં HTMLના વિકાસ, HTML દસ્તાવેજનું માળખું અને દસ્તાવેજ બનાવવા માટે જરૂરી એવાં ટેગનાં ઉદાહરણ સહિત સંપૂર્ણ સમજૂતી આપવામાં આવી છે. સંબંધિત ઉદાહરણ સાથે Head અને Body ટેગની સમજૂતી તેમના ગુણધર્મો સાથે ફરી આપવામાં આવી છે. અંતમાં HTML કોડમાં ચિત્રો, યાદીઓ અને કોષ્ટકનો કેવી રીતે ઉપયોગ થઈ શકે તેની રજૂઆત અને સમજૂતી વિવિધ ઉદાહરણો દ્વારા આપવામાં આવી છે. પ્રકરણ 5 થી 8 માં ઓપનઓફિસના સ્પ્રેડશીટ માટેના ઘટક Calcનો સમાવેશ કરવામાં આવ્યો છે. કેલ્સીના અભ્યાસમાં વિગતોની સુધારણા (editing) અને સંરચના (formatting), કેલ્સી વિધેયો અને આલેખ બનાવવાનો સમાવેશ કરવામાં આવ્યો છે. સમસ્યાના ઉકેલ અને પ્રોગ્રામિંગ અંગેનું પ્રાથમિક જ્ઞાન અત્યંત જરૂરી હોવાને કારણે સી પ્રોગ્રામિંગ ભાષાના ડેટાપ્રકારો, પ્રક્રિયકો, પદાવલિઓ, ઈનપુટ/આઉટપુટ પ્રક્રિયાઓ, નિર્ણયમાળખાં, એરે, લૂપનિયંત્રણ માળખાં અને સી વિધેયોનો સમાવેશ પ્રકરણ 9 થી 16 માં કરવામાં આવ્યો છે. આપેલ મુદ્દાઓનું વિવરણ કરતાં અનેક ઉદાહરણો યોગ્ય જગ્યાએ સમજૂતી સહિત રજૂ કરવામાં આવ્યાં છે.

વિદ્યાર્થીમિત્રો, અપેક્ષિત છે કે જો તમે આ પાઠ્યપુસ્તક અને તેમાં આપેલા પ્રાયોગિક સ્વાધ્યાયનો ધ્યાનપૂર્વક અભ્યાસ કરશો તો કમ્પ્યુટર વિનિયોગોને સમજવા માટે જરૂરી એવા પ્રારંભિક છતાં મહત્વના મુદ્દાઓ જેવા કે HTML, Calc અને C પ્રોગ્રામિંગ સાથેના કાર્યમાં પર્યાપ્ત વિશ્વાસ કેળવી શકશો.





HTMLનો પરિચય

ઇન્ટરનેટ અને HTMLની કાર્યપ્રણાલી

અનેકવિધ વિનિયોગમાં હવે કમ્પ્યુટર બહોળા પ્રમાણમાં ઉપયોગમાં લેવાય છે. જ્યારે બે કે તેથી વધુ કમ્પ્યુટરો એકબીજાં સાથે જોડાય છે ત્યારે તે સંશાધનોને વહેંચી શકે છે. આવાં એકબીજાં સાથે જોડાયેલ કમ્પ્યુટરોના સમૂહને કમ્પ્યુટર નેટવર્ક કહે છે. ઇન્ટરનેટ એ આવાં અનેક કમ્પ્યુટર નેટવર્કનો સમૂહ છે અને તેથી જ તેને અનેક નેટવર્કનું નેટવર્ક કહે છે. ઇન્ટરનેટના પ્લેટફોર્મ પર એક વહેંચાયેલી માહિતી પદ્ધતિ અસ્તિત્વમાં છે, જેને વર્લ્ડ વાઇડ વેબ, WWW અથવા ટૂંકમાં વેબ કહે છે. વેબ અંગેનો પહેલો વિચાર 1991માં ટિમ બર્નર્સ-લી (આકૃતિ 1.1)ને આવ્યો હતો, જ્યારે તેઓ સ્વિટ્ઝરલેન્ડ ખાતે CERN(યુરોપિયન પરમાણુ સંશોધન સંસ્થા, <http://cern.web.cern.ch/CERN/>)માં કામ કરતા હતા. વેબ એ ઇન્ટરનેટ પ્લેટફોર્મ પરની વિવિધ માધ્યમની (મલ્ટીમીડિયા) માહિતીનો ખજાનો છે. વેબપેઇજ સ્વરૂપે ઉપલબ્ધ વેબ-માહિતીને બ્રાઉઝર(વેબમાહિતીને મેળવી નિહાળવા માટેનો ખાસ વિનિયોગ)ના ઉપયોગ દ્વારા જોઈ શકાય છે. આ વેબપેઇજઓમાં વિભિન્ન સ્થાનો તરફ દર્શાવતાં જોડાણો (links) હોય છે. આ જોડાણોને હાઇપરલિન્ક (hyperlink) તરીકે ઓળખવામાં આવે છે. આવી હાઇપરલિન્ક પર ક્લિક કરીને કોઈ પણ વ્યક્તિ પોતાને ઇચ્છિત સ્થાન પર પહોંચાડી શકે છે. આ ક્રિયાને હાઇપરલિન્ક સાથે દોરવાનું કહેવાય છે. શબ્દિક લખાણ, ચિત્ર, આકૃતિ વગેરે કોઈ પણ માહિતીને આવી હાઇપરલિન્ક સાથે સાંકળી શકાય છે. લખાણ અને હાઇપરલિન્કને વેબ પર સીધેસીધા રજૂ કરી શકાય નહીં. વેબ બ્રાઉઝર દ્વારા એક વેબપેઇજ કેવી રીતે દર્શાવાય તે વર્ણવવા માટે HTMLના ટૂંકા નામે ઓળખાતી હાઇપરટેક્સ્ટ માર્કઅપ લેંગ્વેજની જરૂર પડે. આમ, HTML એ વેબપેઇજનું વર્ણન કરતી એક ભાષા છે. HTML એ વેબપેઇજના શીર્ષક, મથાળું, કોષ્ટક, છબી વગેરે જેવા લખાણને નિશાની આપવા માટેની દસ્તાવેજી ભાષા છે. તે યંત્ર સ્વતંત્ર છે અને તેથી HTML સંજ્ઞાના ઉપયોગ વડે લખાયેલ દરેક લખાણને તમામ ઇન્ટરનેટ બ્રાઉઝર સ્વીકારે છે.



આકૃતિ 1.1: ટિમ બર્નર્સ-લી

HTML એ એક પ્રકારની માર્કઅપ (સંજ્ઞાંકિત) ભાષા છે. માર્કઅપ ભાષા એ કેટલીક ચોક્કસ સંજ્ઞાઓ (ટેગ)નો સમૂહ છે, જે વેબના લખાણને કેવી રીતે રજૂ કરવું તે બાબતે વધારાની માહિતી (વેબ માહિતી ઉપરાંત) ધરાવે છે. HTML ફાઇલ એ એવી ટેક્સ્ટ ફાઇલ છે કે જેમાં વેબ-માહિતીની સાથે સાથે ટેગ સ્વરૂપે વધારાની ફોર્મેટિંગ માર્કઅપ માહિતી ધરાવે છે. HTML એ અતિપ્રચલિત માર્કઅપ ભાષા છે અને તે ટેગનો ચોક્કસ સમૂહ ધરાવે છે. HTML એ SGML (Standardized General Markup Language) પરથી ઊતરી આવી છે, જે 1986માં આંતરરાષ્ટ્રીય માનક સંસ્થા (ISO) દ્વારા મશીન વાંચી શકે તેવા દસ્તાવેજોના આદાન-પ્રદાનની સુવિધા માટે તૈયાર કરવામાં આવી હતી.

આમ, HTML કોડ એ બ્રાઉઝર દ્વારા વેબપેઇજ પર દર્શાવવાની માહિતી અને આ માહિતીની રજૂઆતને મદદરૂપ ટેગનું સંયોજન છે. આવા આધારસ્તંભરૂપી કોડ વગર વેબપેઇજ પર માહિતી દર્શાવવાનું અશક્ય છે. આથી વેબપેઇજ તૈયાર કરવા, તેનું અર્થઘટન કરવા અને રજૂઆત કરવા HTML કોડ લખવા ફરજિયાત બને છે.

એક સાદો HTML દસ્તાવેજ

તો ચાલો, HTMLનો ઉપયોગ કરી એક સાદું વેબપેઇજ તૈયાર કરીએ, જે RAINBOW વિશે ચર્ચા કરે. વેબપેઇજ પર રજૂ કરવાની માહિતી કોષ્ટક 1.1માં દર્શાવેલ છે.

RAINBOW

Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

કોષ્ટક 1.1 : HTML વડે વેબપેઈજ પર રજૂ કરવાનું લખાણ

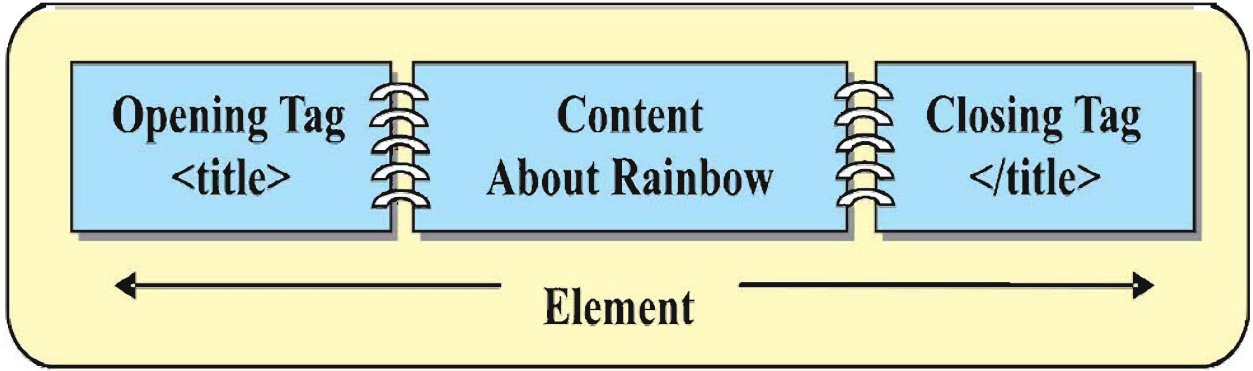
કોષ્ટક 1.1માં દર્શાવેલ લખાણને દર્શાવવા માટેના HTML કોડ કોડ-લિસ્ટિંગ 1.1માં આપેલ છે.

```
<html>
  <head>
    <title> About Rainbow
  </title>
</head>
<body>
  <h1> RAINBOW </h1>
  <p>  Rainbow consists of seven colours. These colours
    are Violet, Indigo, Blue, Green, Yellow, Orange and
    Red. They are also acronymed as VIBGYOR.
  </p>
  <p>  Rainbow is caused by reflection of light in water
    droplets in the Earth's atmosphere, resulting in a
    spectrum of light appearing in the sky. It takes the
    form of a multi coloured arc.
  </p>
</body>
</html>
```

કોડ-લિસ્ટિંગ 1.1

અહીં એ ખાસ અવલોકન કરો કે કોડ-લિસ્ટિંગ 1.1માં rainbowના લખાણ ઉપરાંત કેટલાક ખૂણિયા કૌંસ અને તે દરેકમાં કોઈ ચોક્કસ અક્ષર કે શબ્દો પણ દર્શાવાયા છે. આ ખૂણિયા કૌંસ અને તેની અંદરના શબ્દોને ટેગ કહે છે. ટેગ એ બે બાજુ ખૂણિયા કૌંસથી આવરેલ અક્ષરો, શબ્દો અને આંકડાઓનો બનેલો હોય છે. શરૂ થતા ટેગ અને બંધ થતા ટેગની વચ્ચે જરૂરી લખાણને એકસાથે એક એલિમેન્ટ (Element) કહે છે. HTML એલિમેન્ટ ખાલી હોય

અથવા માહિતી દર્શાવવા માટે ગોઠવણી અને પ્રકાશન સંબંધિત વધારાની સૂચનાઓ પણ હોઈ શકે. આકૃતિ 1.2 ઉદાહરણ સાથે ટેગનું બંધારણ અને એલિમેન્ટ દર્શાવે છે.



આકૃતિ 1.2 : HTML એલિમેન્ટ

આકૃતિ 1.2 એવું દર્શાવે છે કે, **<title> About Rainbow </title>** એ HTML કોડનો એલિમેન્ટ છે. અહીં એ નોંધ લો કે બંધ થતા ટેગમાં શબ્દની આગળ સ્લેશ **</title>** મૂકવામાં આવ્યો છે. આ ટેગ એલિમેન્ટનો અંત દર્શાવે છે. ટેગની જોડ અને તેની વચ્ચેનું લખાણ લેગું મળીને, HTML એલિમેન્ટ બનાવે છે. અહીં Title ટેગ HTML કોડ વડે બનતા વેબપેઈજનું શીર્ષક નક્કી કરે છે.

આવા અન્ય એક HTML એલિમેન્ટનું ઉદાહરણ નીચે મુજબ છે :

<h1> RAINBOW </h1>

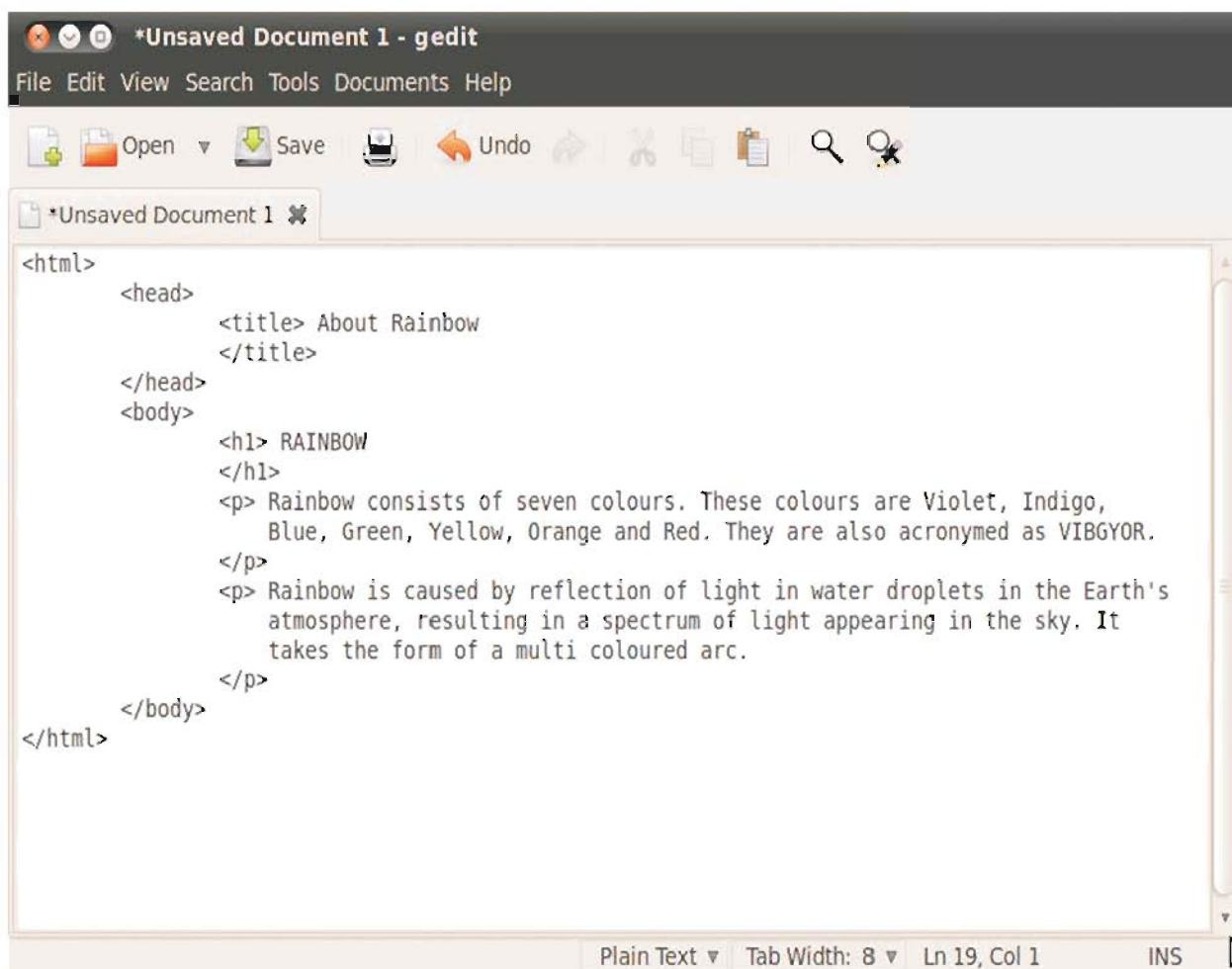
અહીં h1 ટેગ મથાળા માટે છે.

ઉપર દર્શાવ્યા મુજબ **<h1>** અને **</h1>** ની વચ્ચેનું લખાણ મથાળા તરીકે ઓળખવામાં આવશે અને મથાળા તરીકે રજૂ કરવામાં આવશે. એ જ રીતે, **<p>** અને **</p>** ટેગ વચ્ચે ઉપલબ્ધ લખાણ ફકરા તરીકે ગણવામાં આવશે અને જુદા ફકરા તરીકે રજૂ કરવામાં આવશે. આખો દસ્તાવેજ **<html>** અને **</html>** ટેગમાં આવરીને દર્શાવાશે.

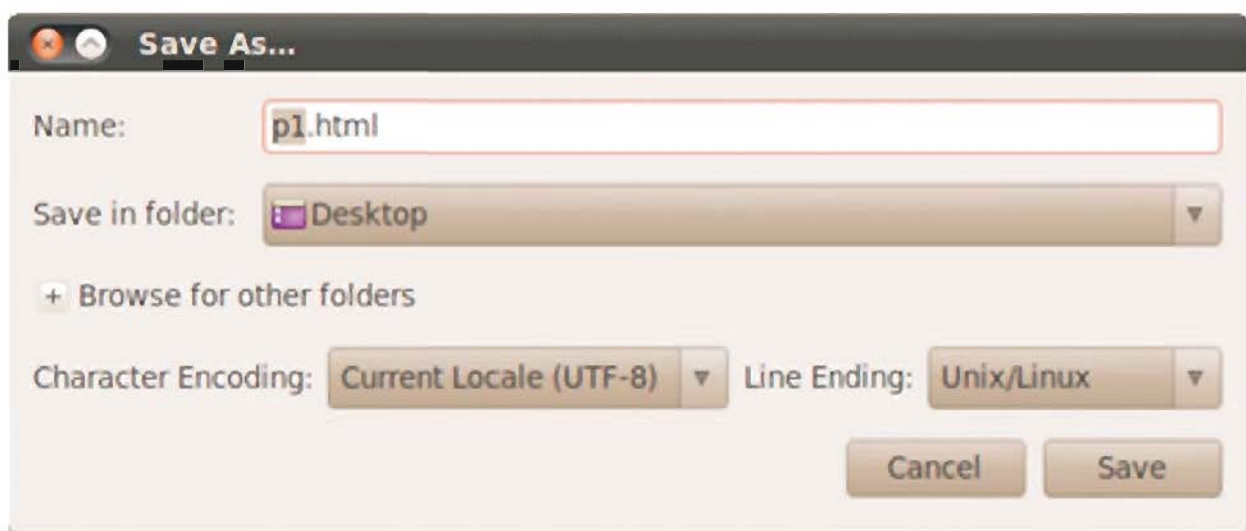
બ્રાઉઝર દ્વારા આ પાનું કેવું દર્શાવાશે તે નિહાળવા નીચે આપેલાં પગલાં અનુસરો :

પગલું 1 : Applications → Accessories → gedit પસંદ કરી gedit નામનો એડિટર પ્રોગ્રામ ખોલો. gedit એ GNOME (GNU નામે ઓળખાતા પ્રોજેક્ટનો ભાગ, MIT દ્વારા બનાવાયેલ મુક્ત સોફ્ટવેર), ડેસ્કટોપ એન્વાયર્નમેન્ટ, Mac OS X અને માઈક્રોસોફ્ટ વિન્ડોઝ માટેનું સામાન્ય હેતુ માટેનું ટેક્સ્ટ એડિટર છે. એના બદલે, તમે સ્ક્રીનને મથાળે ઉપલબ્ધ હરોળમાં gedit એડિટરના શોર્ટકટનો ઉપયોગ પણ કરી શકો.

પગલું 2 : ખાલી gedit વિન્ડોમાં કોડ-લિસ્ટિંગ 1.1માં આપેલ HTML કોડ-લિસ્ટિંગ ટાઈપ કરો. આ કોડ ટાઈપ કરી દીધા પછીનો gedit એડિટરનો દેખાવ આકૃતિ 1.3માં દર્શાવેલ છે. એડિટરના મથાળાની ઘાઈન પર આપેલ Save વિકલ્પ પસંદ કરીને "p1.html" તરીકે સંગ્રહ કરો. અહીં એ બાબતની નોંધ લો કે, HTML ફાઈલ htm અથવા html અનુલંબન સાથે સાચવી શકાય છે. આકૃતિ 1.4 કોડને સાચવવા માટેની રીત દર્શાવે છે.



આકૃતિ 1.3 : gedit એડિટરમાં લખેલ HTML કોડ



આકૃતિ 1.4 : Save As... ડાયલોગ બોક્સ

પગલું 3 : Mozilla Firefox કે તમારા કમ્પ્યુટરમાં પ્રસ્થાપિત કરેલ અન્ય કોઈ પણ બ્રાઉઝરને ખોલો. File → Open File પસંદ કરો. તમને Open File ડાયલોગ બોક્સ દર્શાવાશે. તમારે જે ફાઈલને ખોલવી હોય તે પસંદ કરી Open બટન પર ક્લિક કરો. એના વિકલ્પરૂપે તમે ફાઈલ પર ડબલ ક્લિક પણ કરી શકો. ફાઈલને જો Mozilla Firefoxમાં ખોલવામાં આવે તો તેનો દેખાવ અને અસર આકૃતિ 1.5 દર્શાવે છે.



RAINBOW

Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

Done

આકૃતિ 1.5 : HTML કોડનો ઉપયોગ કરી બ્રાઉઝર દ્વારા દર્શાવાતું વેબપેઈજ

અત્રે એ ખાસ નોંધવું જોઈએ કે, HTML કોડમાં ઉપયોગમાં લેવાતા ટેગ કેસ-સેન્સિટિવ નથી. એટલે કે, આપણે ટેગને મોટા (અપર કેસ) કે નાના (લોઅર કેસ) અક્ષરોમાં અથવા બન્નેનું મિશ્રણ કરીને પણ લખી શકીએ. વધુમાં, એકથી વધુ એલિમેન્ટ એક જ લીટીમાં પણ લખી શકાય છે. જોકે, બને ત્યાં સુધી દરેક એલિમેન્ટને નવી લીટી પર લખવા જોઈએ અને વાંચવામાં સરળતા રહે તે માટે યોગ્ય ઈન્ડેન્ટ પાડીને લખવું જોઈએ.

HTML દસ્તાવેજનું માળખું

HTML દસ્તાવેજ મુખ્યત્વે બે ભાગમાં વહેંચાયેલું હોય છે : પ્રથમ મથાળા વિભાગ (Head section) હોય છે અને બીજો બોડી વિભાગ (Body section). તેને ક્યારેક હેડ એલિમેન્ટ (Head Element) અને બોડી એલિમેન્ટ (Body Element) તરીકે પણ ઓળખવામાં આવે છે. મથાળા વિભાગમાં પાના વિશેની માહિતી જેવી કે, શીર્ષક (title) અને પાનાનું વર્ણન (description) ધરાવે છે. આ બધી જ માહિતી <head> અને </head> ટેગની વચ્ચે આવરીને દર્શાવવી પડે છે.

બોડી વિભાગ <body> અને </body> ટેગની વચ્ચે આવરીને દર્શાવાય છે. આ એ વિગત છે જે બ્રાઉઝર મારફત જોઈ શકાય છે. મથાળા વિભાગ અને બોડી વિભાગ એ બન્ને <html> અને </html> ટેગની વચ્ચે આવરીને દર્શાવાય છે.

HTML શીર્ષક

વેબપેઈજનું શીર્ષક TITLE એલિમેન્ટ દ્વારા રજૂ કરાય છે, જે દસ્તાવેજના મથાળા વિભાગમાં મૂકવામાં આવે છે. અત્રે એ ખાસ નોંધનીય છે કે કોઈ પણ દસ્તાવેજને માત્ર એક જ title એલિમેન્ટ હોય છે. સામાન્ય રીતે દસ્તાવેજના લખાણને ઓળખવા માટે તે ઉપયોગી છે. વળી, titleનું લખાણ દસ્તાવેજના લખાણનો ભાગ નથી. આ કારણોસર આ લખાણ સાદું લખાણ હોવું જોઈએ અને તેમાં હાઈપરલિન્ક જેવા વિશેષ કમાન્ડ ધરાવતા હોવા ન જોઈએ. આ Title એ લખાણને દર્શાવતી વિન્ડોના શીર્ષક તરીકે દેખાશે. Bookmark list અથવા બ્રાઉઝરના ઇતિહાસ(history)માં પણ આ title નું સ્થાન રહે છે. અને તેથી title બને તેટલાં ટૂંકાં હોવાં જોઈએ તેવી ભલામણ કરવામાં આવે છે. કોડ-લિસ્ટિંગ 1.1ના કોડ-લિસ્ટિંગ મુજબ આપણા ઉદાહરણના HTML કોડ માટે "About Rainbow" એ શીર્ષક છે. આકૃતિ 1.5માં દર્શાવ્યા મુજબ તે દર્શાવતી વિન્ડોના ઉપરના ભાગે દર્શાવાય છે.

HTML હેડિંગ સ્ટાઈલ

HTML દસ્તાવેજ સામાન્ય રીતે મથાળા સાથે શરૂ થાય છે. કોડ-લિસ્ટિંગ 1.1માં દર્શાવેલ ઉદાહરણ મુજબ મથાળા માટે પ્રથમ પ્રકારની સ્ટાઈલ (h1) ટેગ ઉપયોગમાં લેવાઈ છે. મથાળાની પ્રથમ પ્રકારની સ્ટાઈલ (h1) કેવી દેખાય છે તે જોવા માટે આકૃતિ 1.5માં દર્શાવેલ મથાળા RAINBOWને જુઓ. HTMLમાં આવી વધુ પાંચ પ્રકારની મથાળાની સ્ટાઈલ

ઉપલબ્ધ છે. કુલ છ પૂર્વગ્રસ્થાપિત સ્ટાઈલ અનુસાર મથાળું દર્શાવી શકાય છે, જે અનુક્રમે h1, h2, h3, h4, h5 અને h6 નામથી ઓળખાય છે. મથાળાના આ છ સ્તર આકૃતિ 1.6માં આપેલ HTML કોડમાં વર્ણવેલ છે.



```

Source of: file:///home/pss/Desktop/p2.html - Mozilla Firefox
File Edit View Help

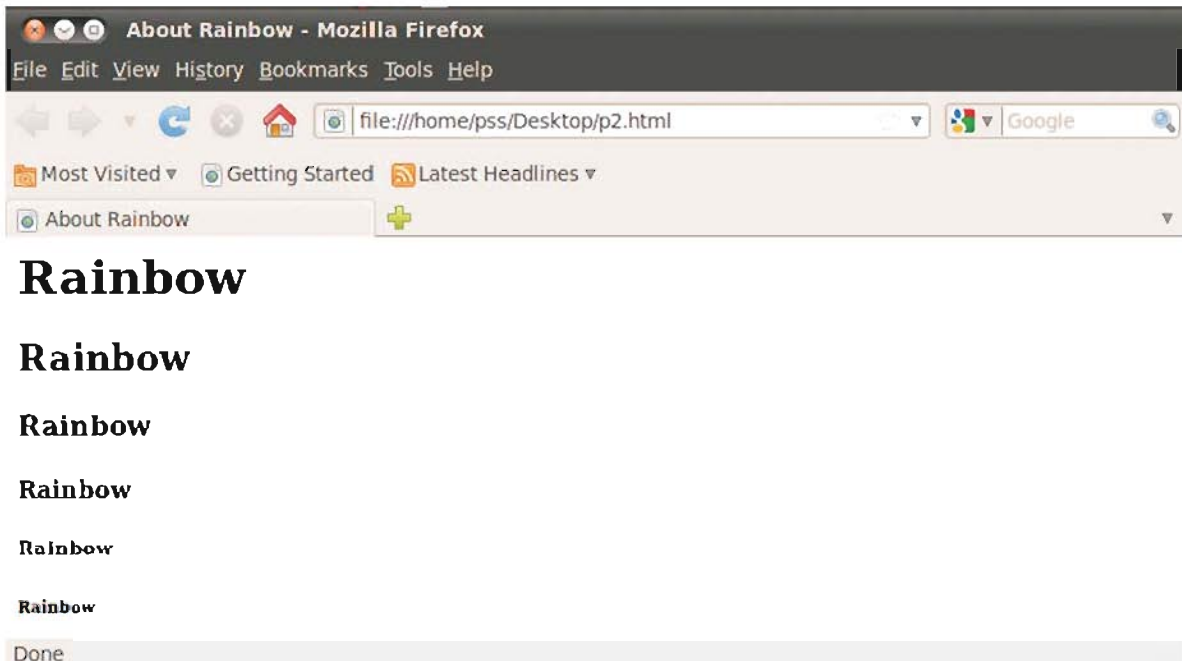
<html>
  <head>
    <title> About Rainbow
  </title>
  </head>
  <body>
    <h1> Rainbow </h1>
    <h2> Rainbow </h2>
    <h3> Rainbow </h3>
    <h4> Rainbow </h4>
    <h5> Rainbow </h5>
    <h6> Rainbow </h6>
  </body>
</html>

```

આકૃતિ 1.6 : HTMLમાં જુદા જુદા સ્તરના મથાળા માટેનો HTML કોડ

gedit એડિટરનો ઉપયોગ કરીને આ કોડ ટાઇપ કરવામાં આવ્યા છે. આ કોડનો "p2.html" નામથી સંગ્રહ કરો. જ્યારે આ કોડ આપણે બ્રાઉઝર દ્વારા નિહાળીશું ત્યારે તે આકૃતિ 1.7માં દર્શાવ્યા મુજબનું દૃશ્ય રજૂ કરશે.

મોટા ભાગનાં બ્રાઉઝર <h1>, <h2> અને <h3>નાં લખાણને દસ્તાવેજના લખાણના સામાન્ય કદ કરતાં મોટા કદમાં દર્શાવે છે. <h4> એલિમેન્ટના લખાણનું કદ લખાણના સામાન્ય કદ જેટલું જ દર્શાવાશે. જોકે, કોઈ વ્યક્તિ આ મથાળાના કદને હંમેશાં પોતાની રીતે પુનઃનિર્ધારિત કરી શકે છે.



આકૃતિ 1.7 : HTMLમાં મથાળાના જુદા જુદા સ્તરનું પરિણામ

લખાણની સુઘડતા વધારતા ટેગ

h1 થી h6 જેવી મથાળાની વિવિધ સ્ટાઇલ ઉપરાંત HTMLમાં બીજી ઘણી એવી ટેગ લખાણના ફોર્મેટિંગ માટે ઉપલબ્ધ છે. આ ટેગને HTML કોડના બોડી વિભાગમાં ઉપયોગમાં લઈ શકાય છે. આવા કેટલાક મૂળભૂત ટેગ નીચે મુજબ છે :

ફકરા માટેના <p> અને </p> ટેગ

<p> ટેગ લખાણને ફકરા સ્વરૂપે ગોઠવે છે. લખાણના દરેક ફકરાને <p> અને </p> ટેગની વચ્ચે મૂકવો જોઈએ. નીચે આવાં ત્રણ યોગ્ય ઉદાહરણ આપેલ છે :

<p> This is first paragraph. </p>

<p> The second paragraph is here. This is about multiple colours of rainbows.

Though it is a temporary event it leaves a great impact on human mind. </p>

<p> Best of Luck! </p>

ઉપર દર્શાવેલ ઉદાહરણો પૈકી બીજું ઉદાહરણ ધ્યાને લો. તમે એ જોઈ શકો છો કે, બીજા ફકરામાં અનેક ખાલી જગ્યા અને ક્રેજ રિટર્ન (જ્યાં Enter કી દબાવી છે અને લખાણ નવી લીટી પર આવી ગયું છે.) જણાય છે. અહીં એ નોંધનીય છે કે, આવી ખાલી જગ્યા(white space)ને એક જ ખાલી જગ્યા તરીકે ગણવામાં આવશે. જ્યારે HTML કોડ બ્રાઉઝરમાં દર્શાવાશે ત્યારે લખાણ આપમેળે સ્ક્રીનની સંપૂર્ણ પહોળાઈ લઈ લેશે. જો તમે બ્રાઉઝરની વિન્ડોનું કદ બદલશો તો બ્રાઉઝર લખાણને નવી લાઈનમાં લઈ જશે. આવી ખાલી જગ્યા (white space)ની વ્યવસ્થા HTML કોડ તૈયાર કરનારને લખેલ કોડ સારી રીતે વાંચી શકાય તે માટે કોડને ઇન્ડેન્ટ પાડીને દર્શાવવાની તેમજ કોડની વચ્ચે જરૂરી વધારાની ખાલી જગ્યા ઉમેરવાની છૂટ આપે છે.

નવી લીટી પરથી શરૂઆત માટે:
 અથવા

અન્ય ટેગ કરતાં ઊલટું,
 એલિમેન્ટને શરૂઆતનો અને અંતનો ટેગ હોતો નથી.
 એ breakનું ટૂંકું નામ છે. આવા ટેગ ખાલી (Empty) ટેગ તરીકે પણ ઓળખાય છે. XHTML જેવી અદ્યતન આવૃત્તિ
 ટેગનો ઉપયોગ કરે છે. અહીં એ નોંધ લો કે, 'br' અને '/'ની વચ્ચે જગ્યા હોવી જરૂરી છે.

 તેના પછી આવતા લખાણને નીચેની લીટી પર લઈ જાય છે. જો તમારે એકથી વધુ ખાલી લીટીઓ છોડવી હોય તો જરૂર મુજબ એકથી વધુ
 ટેગ મૂકો.
નો ઉપયોગ દર્શાવતું ઉદાહરણ નીચે મુજબ છે :

First example

Second example

ફકરામાં
નો ઉપયોગ કરવાથી લખાણમાં ફરજિયાત બ્રેક લાવે છે, અને રજૂઆત વખતે તેમજ બ્રાઉઝર વિન્ડોનું કદ બદલતી વખતે લખાણને વાળીને (wrap કરીને) નવી લાઈન પર લઈ જવામાં તકલીફ પડે છે.

પ્રિ-ફોર્મેટેડ લખાણ

ઘણી વાર આપણે આપણા લખાણને બ્રાઉઝર બદલે નહીં તેવી રીતે એકથી વધુ ખાલી જગ્યા (white space) સાથે એકથી વધુ લીટીઓમાં દર્શાવાય એવું ઇચ્છીએ છીએ. આ માટે આપણે આપણા આવા લખાણને <pre> અને </pre> નામના બે પ્રિ-ફોર્મેટેડ ટેગમાં આવરીને દર્શાવી શકીએ. <pre> સાથે શરૂ થતાં અને </pre> સાથે પૂરા થતા કોઈ પણ લખાણની ગોઠવણી જેમની તેમ જળવાઈ રહે છે. ઉદાહરણ નીચે મુજબ છે :

<pre>

This is first line.

This is second line.

This is third line.

</pre>

બ્રાઉઝરમાં દર્શાવાતું લખાણ આપેલ તમામ ઇન્ડેન્ટ (indentation) સાથે દેખાય છે. નીચેનાં ઉદાહરણનો પણ પ્રયોગ કરી જુઓ :

**Bold: and **

નક્કી કરેલ લખાણને ઘાટા અક્ષરોમાં છાપવા માટે આ ટેગનો ઉપયોગ કરાય છે. આનું ઉદાહરણ નીચે મુજબ રજૂ કરી શકાય :

<p> This is the first paragraph. </p>

અહીં "first" શબ્દને ઘાટા અક્ષરોમાં દર્શાવવામાં આવશે.

Underline: <u> અને </u>

લખાણમાંના અમુક અગત્યના શબ્દ નીચે લીટી દોરેલી દર્શાવાય તે માટે આ ટેગ ઉપયોગમાં લઈ શકાય છે, જેનું ઉદાહરણ નીચે મુજબ આપી શકાય :

<p> This is the <u> first </u> paragraph. </p>

અહીં, first શબ્દની નીચે લીટી સાથે દર્શાવાશે.

Italics: <i> and </i>

લખાણમાંના અમુક શબ્દોને ત્રાંસા (ઇટાલિક્સ ઢબે) છાપવા માટે આ ટેગનો ઉપયોગ કરવામાં આવે છે, જેનું ઉદાહરણ નીચે મુજબ આપી શકાય :

<p> This is the <i> first </i> paragraph. </p>

અહીં first શબ્દ ત્રાંસા અક્ષરો સાથે છપાશે.

છેકી નાંખેલા શબ્દો: <s> અને </s>

<s> અથવા <strike> એલિમેન્ટના લખાણના શબ્દો લખ્યા પછી છેકી નાંખ્યા હોય તે રીતે શબ્દો પર પાતળી લીટી સાથે છપાય છે. અહીં s એ strike શબ્દનું ટૂંકું નામ છે. છેકી નાંખેલા શબ્દોને રજૂ કરતું ઉદાહરણ નીચે મુજબ છે :

<p> This is the <s> cancelled </s> paragraph. </p>

અહીં cancelled લખાણ છેકી નાંખેલા શબ્દો જેવું છપાશે.

ટાઇપરાઈટર ફોન્ટ: <tt> અને </tt>

<tt> એલિમેન્ટનું લખાણ ટાઇપરાઈટર જેવા અક્ષરોમાં લખાય છે, જેને 'મોનો-સ્પેસ ફોન્ટ' (ટેલિ-ટાઇપ મશીન જેવા) તરીકે પણ ઓળખાય છે, જેનું ઉદાહરણ નીચે મુજબ છે :

<p> This is the <tt> first </tt> paragraph. </p>

અહીં first શબ્દ મોનો-સ્પેસ ફોન્ટમાં દર્શાવાશે.

આવા કેટલાક અન્ય એલિમેન્ટ પણ છે, જે કોષ્ટક 1.2માં દર્શાવેલ છે.

એલિમેન્ટ	વર્ણન
<small> અને </small>	આ એલિમેન્ટનું લખાણ તેની આસપાસના લખાણ કરતાં એક ફોન્ટ સાઈઝ જેટલું નાનું દર્શાવાય છે.
<big> અને </big>	આ એલિમેન્ટનું લખાણ તેની આસપાસના લખાણ કરતાં એક ફોન્ટ સાઈઝ જેટલું મોટું દર્શાવાય છે.
^{અને}	આ એલિમેન્ટનું લખાણ તેની આગળના લખાણ કરતાં નાનું અને તેના કરતાં થોડું ઊંચે (સુપરસ્ક્રિપ્ટ અક્ષર તરીકે) છપાય છે.
_{અને}	<sup> કરતાં ઊલટું, આ એલિમેન્ટનું લખાણ તેની આગળના લખાણ કરતાં નાનું અને તેના કરતાં થોડું નીચે (સબસ્ક્રિપ્ટ અક્ષર તરીકે) છપાય છે.
<acronym> અને </acronym>	આ એલિમેન્ટનું લખાણ ટૂંકા નામ (acronym) તરીકે દર્શાવાય છે.
<dfn> અને </dfn>	તે એક ચોક્કસ પદને વ્યાખ્યાયિત કરે છે.
<q> અને </q>	તે અવતરણ ચિહ્ન (quote) દર્શાવે છે.

કોષ્ટક 1.2 : કેટલાંક અન્ય ફોર્મેટિંગ ટેગ

Anchor Tag

જ્યારે કોઈ લખાણ HTML દસ્તાવેજની અંદર જ દર્શાવાય છે, ત્યારે લખાણ અને તેના સ્વરૂપને નિયંત્રિત કરતા અક્ષરો (ટેગ) ઉપરાંત અન્ય લખાણને સંબોધતી વધારાની માહિતી પણ જરૂરી બને છે. ઘણી વખત વિશેષ ખુલાસો પણ જરૂરી હોય છે. આવા કેટલાક અક્ષરો કે શબ્દોનો સમૂહ કે જે અલગ રંગથી (સામાન્ય રીતે ભૂરા રંગથી) અને નીચે લીટી દોરીને દર્શાવાય છે તેને હાઈપરલિન્ક તરીકે ઓળખાય છે. હાઈપરલિન્ક <a> નામના ટેગ દ્વારા બનાવવામાં આવે છે, જેમાં 'a'નો અર્થ anchor છે. તો આવો, "p1.html"ને કોડ-લિસ્ટિંગ 1.2માં બતાવ્યા પ્રમાણે સુધારીએ.

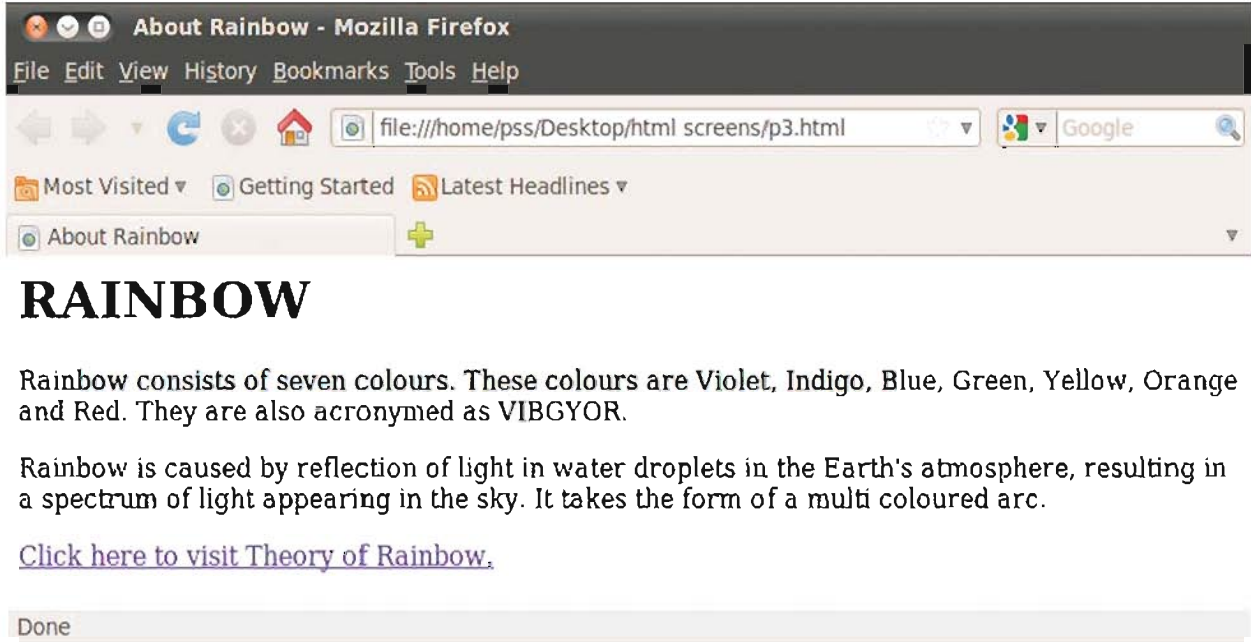
```

<html>
  <head>
    <title> About Rainbow
  </title>
  </head>
  <body>
    <h1> RAINBOW </h1>
    <p> Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.
    </p>
    <p> Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.
    </p>
    <p>
      <a href= "p4.html" > Click here to visit Theory of Rainbow. </a>
    </p>
  </body>
</html>

```

કોડ-લિસ્ટિંગ 1.2 : હાઈપરલિન્કનો ઉપયોગ દર્શાવતો HTML કોડ

ફાઈલને "p3.html" તરીકે સાચવો. જ્યારે આ કોડને બ્રાઉઝર દ્વારા નિહાળવામાં આવશે ત્યારે તે આકૃતિ 1.8 મુજબ દર્શાવાશે.



આકૃતિ 1.8 : p3.htmlનું પરિણામ

જ્યારે ઉપયોગકર્તા હાઈપરલિન્ક પર ક્લિક કરે ત્યારે 'Theory of Rainbow' વર્ણવતી નિર્ધારિત ફાઈલ ખુલવી જોઈએ. તો ચાલો, હાઈપરલિન્ક દ્વારા નિર્દિષ્ટ ફાઈલ માટે કોડલિસ્ટિંગ 1.3માં દર્શાવ્યા મુજબ HTML કોડ બનાવીએ.

```
<html>

  <head>

    <title> Theory of Rainbow

  </title>

  </head>

  <body>

    <h1> How Rainbow Developed </h1>

    <p> Rainbow is caused by reflection of light in water droplets
      in the Earth's atmosphere, resulting in a spectrum of light
      appearing in the sky. It takes the form of multi-coloured arc.

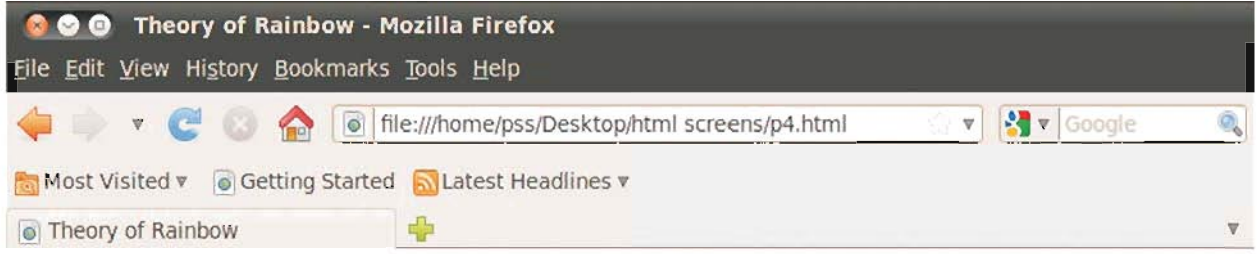
    </p>

  </body>

</html>
```

કોડ-લિસ્ટિંગ 1.3 : હાઈપરલિન્ક સાથે જોડાયેલ માહિતી માટેનો HTML કોડ

કોડ-લિસ્ટિંગ 1.3માં દર્શાવેલ HTML કોડનો "p4.html" તરીકે સંગ્રહ કરો. જ્યારે તમે આકૃતિ 1.8માં દર્શાવ્યા મુજબના સાંકળેલ (anchor) લખાણ (Click here to visit Theory of Rainbow) પર ક્લિક કરશો તો તરત જ p4.html ફાઈલનું લખાણ બ્રાઉઝરમાં આવી જશે. (જુઓ આકૃતિ 1.9.)



How Rainbow Developed

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

Done

આકૃતિ 1.9 : p4.htmlનું લખાણ

હાઈપરલિન્ક એ દસ્તાવેજમાં હાઈપર ટેક્સ્ટ (hyper text) બનાવીને તેને અન્ય દસ્તાવેજ સાથે સાંકળીને બે HTML દસ્તાવેજને જોડવાની રીત છે. એક વેબસાઈટ, કે જે ઘણાં બધાં વેબપેઈજનો સમૂહ હોય છે, તે આવા હાઈપરલિન્ક સંચાલન મારફત સંચાલિત થાય છે. અહીં એ નોંધનીય છે કે, આપણે બનાવેલાં વેબપેઈજ, તેના HTML કોડમાં આપણે જે રીતે દર્શાવેલ છે તે પ્રમાણે લખાણને રજૂ કરે છે. આવાં વેબપેઈજ દર્શાવેલ માહિતીને નિયત (static કે fixed) સ્વરૂપે જ રજૂ કરતા હોઈ તેને “સ્ટેટિક વેબપેજ” તરીકે ઓળખવામાં આવે છે.

પૂર્ણ અને સંબંધિત સરનામું

કોડ-લિસ્ટિંગ 1.2માં દર્શાવેલ ` Click here to visit Theory of Rainbow. ` ને જુઓ.

`http://www.somedomain.com/p4.html` એવું પૂર્ણ સરનામું આપવાને બદલે આપણે માત્ર ફાઈલનું નામ "p4.html" જ આપ્યું છે. આ રીતે માત્ર ફાઈલનું નામ જ દર્શાવવાનું ત્યારે જ શક્ય છે કે જ્યારે આપણી p3.html ફાઈલ કે જેમાંથી અન્યને બોલાવવામાં આવે છે તે. (આવી ફાઈલને ‘કોલિંગ ફાઈલ’ કે ‘પેરન્ટ ફાઈલ’ કહે છે.) અને "p4.html" જેને બોલાવવામાં (call કરવામાં) આવે છે, (આવી ફાઈલને ‘રીફર્ડ ફાઈલ’ કે ‘કોલ ફાઈલ’ કહે છે.) તે બન્ને એક જ રિસ્કેટરીમાં ઉપલબ્ધ હોઈ બોલાવવામાં આવતી (called) ફાઈલનું સ્થાન એ બોલાવનાર ફાઈલને સંબંધિત હોય છે અને એ રીતે એને સંબંધિત સરનામા (Relative Address) તરીકે ઓળખાય છે. જ્યારે સંપૂર્ણ સરનામાને પૂર્ણ સરનામા (absolute address) તરીકે ઓળખવામાં આવે છે. જો બોલાવવામાં આવતી ફાઈલ એક રિસ્કેટરી ઉપર આવેલી હોય તો આપણે ફાઈલ નામની આગળ ../ લગાવી શકીએ.

નોંધ :

જો ફાઈલના નામની આગળ કોઈ સરનામું આપવામાં ન આવ્યું હોય તો બ્રાઉઝર એવું માની લેશે કે બોલાવવામાં આવેલી ફાઈલ બોલાવનાર (પેરન્ટ) ફાઈલની રિસ્કેટરીમાં જ છે.

ટેગના ગુણધર્મો

ટેગની સાથે વધુ માહિતીને દર્શાવવા ટેગની સાથે વધારાના ગુણધર્મો જોડવામાં આવે છે. બીજા શબ્દોમાં કહીએ તો વિવિધ ગુણધર્મો (એલિમેન્ટ) બાબતે વધુ વિગતો આપે છે. કોઈ પણ એલિમેન્ટના ગુણધર્મો તેના શરૂઆતના (ઓપનિંગ) ટેગની સાથે દર્શાવાય છે. ગુણધર્મ બે ભાગના બનેલા હોય છે, જેનો પહેલો ભાગ ગુણધર્મનું નામ હોય છે અને બીજો ભાગ તેના માટે જરૂરી કિંમતનો હોય છે.

ગુણધર્મનું નામ ક્યો ગુણધર્મ ગોઠવવો તે દર્શાવે છે. કોડ-લિસ્ટિંગ 1.2માં દર્શાવેલ <a> ટેગમાં href એ ગુણધર્મનું નામ છે. કિંમત એ ગુણધર્મ માટે ગોઠવવાની કિંમત છે. href/ના કિસ્સામાં કિંમત p4.html છે (જે જોડાણનું સરનામું છે). આ કિંમત બે અવતરણ ચિહ્નમાં આવરીને દર્શાવવી પડે. નામ અને કિંમતની વચ્ચે બરાબર (=) નું ચિહ્ન મૂકવું પડે છે. ગુણધર્મ સાથેની ટેગ કેવી રીતે દર્શાવવી તેનાં ઉદાહરણ હવે પછીના વિભાગમાં જોઈએ.

Align ગુણધર્મ

align ગુણધર્મ એવું દર્શાવે છે કે, મથાળું પાનાની ડાબી બાજુ, જમણી બાજુ કે મધ્યમાં કયા સ્થાને દેખાવું જોઈએ. સામાન્ય રીતે લખાણ પાનાની ડાબી બાજુ જ ગોઠવાતું હોય છે. Align ગુણધર્મ સાથે નીચેની ત્રણ પૈકીની કોઈ એક કિંમત ગોઠવી શકાય.

Left : લખાણ પાનાની ડાબી બાજુ ગોઠવાય છે.

Right : લખાણ પાનાની જમણી બાજુ ગોઠવાય છે.

Center : લખાણ પાનાની મધ્યમાં ગોઠવાય છે.

align ગુણધર્મના ઉપયોગો દર્શાવતાં કેટલાંક ઉદાહરણો નીચે મુજબ છે :

<p align="right"> This content will be displayed in right aligned form </p>

<p align="center"> This content will be displayed in center position of the page </p>

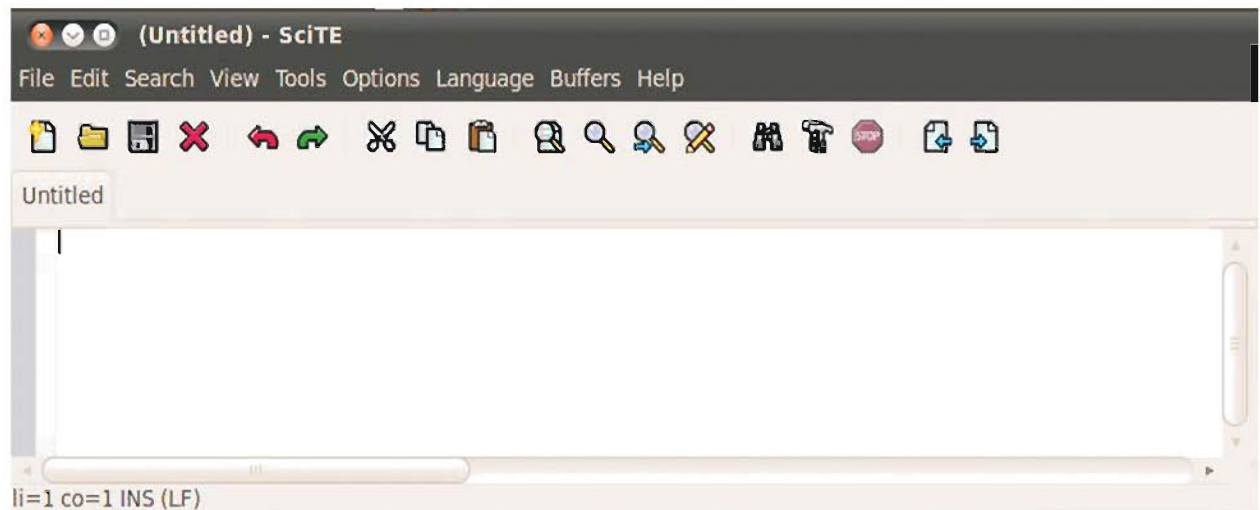
એવા પણ કેટલાક ગુણધર્મો છે કે જે લગભગ દરેક ટેગ સાથે આવી શકે છે. આવા ગુણધર્મને સાર્વત્રિક ગુણધર્મ કહેવાય. align એ આવો જ એક ગુણધર્મ છે. સાર્વત્રિક ગુણધર્મ હોવાને લીધે align ગુણધર્મને નીચે દર્શાવ્યા મુજબ heading સાથે પણ વાપરી શકાય.

<h1 align="center"> Centered Heading </h1>

જ્યારે ફકરાના લખાણને સરખું ગોઠવવામાં (align) આવે છે, ત્યારે સરખી ગોઠવણ કરવા માટે લખાણમાં શબ્દોની વચ્ચે ક્યાંક ક્યાંક વધારાની ખાલી જગ્યા ઉમેરવામાં આવે છે. આ રીતે આપમેળે ઉમેરાતી ખાલી જગ્યાને “સોફ્ટ સ્પેસ” (soft space) કહે છે. જો ઉપયોગકર્તા પોતે ખાલી જગ્યા ઉમેરે તો તેને “હાર્ડ સ્પેસ” (hard space) કહેવાય. જો HTML કોડમાં <pre> અને </pre> ટેગની વચ્ચે લખાયેલ લખાણ સિવાયના લખાણમાં આવી હાર્ડ સ્પેસ હશે તો તે આપમેળે કાઢી નાંખવામાં આવશે.

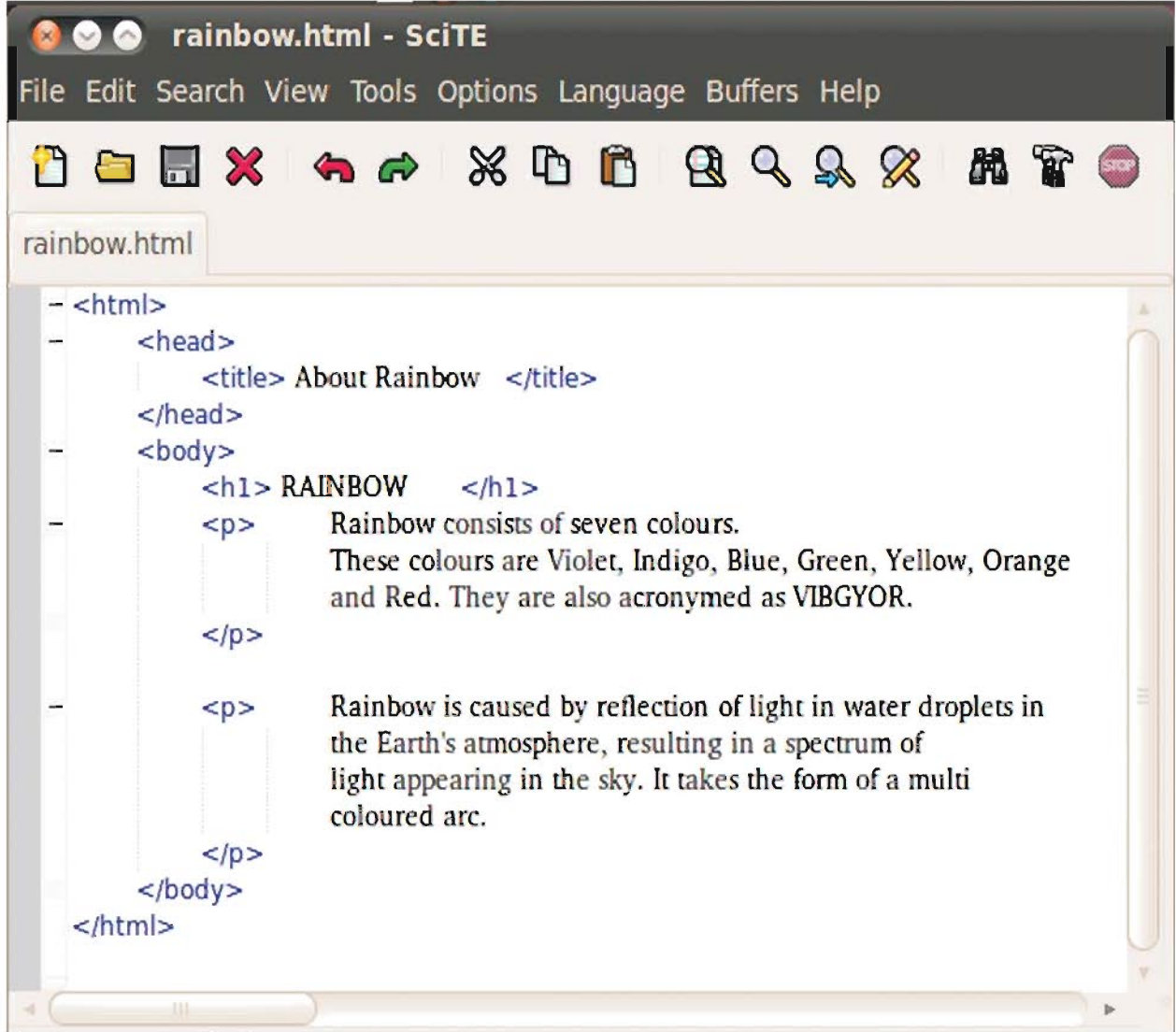
અન્ય એડિટર SciTEનો ઉપયોગ

SciTE એ લખાણને સુધારવા માટેનું એક ટેક્સ્ટ એડિટર (Text Editor) છે, જે Scintilla [http://www.scintilla.org] નામના “ફ્રી સોર્સ કોડ એડિટિંગ કમ્પોનન્ટ” (Free Source Code Editing Component) પર આધારિત છે. તે પૂરેપૂરા સોર્સ કોડ (source code) અને પરવાના સાથે આવે છે, જે તેનો કોઈ પણ નિઃશુલ્ક કે ધંધાકીય સોફ્ટવેર ઉત્પાદન માટે ઉપયોગ કરવાની અનુમતિ આપે છે. SciTE નો દેખાવ આકૃતિ 1.10માં દર્શાવ્યા મુજબનો દેખાય છે.



આકૃતિ 1.10 : SciTE એડિટરનો દેખાવ

અગાઉ મેઘધનુષ્ય (rainbow) વિશેની માહિતી દર્શાવવા માટે લખાયેલ HTML કોડ(જુઓ આકૃતિ 1.3.)ને આકૃતિ 1.11માં દર્શાવ્યા મુજબ SciTE એડિટરમાં પણ લખી શકાય.



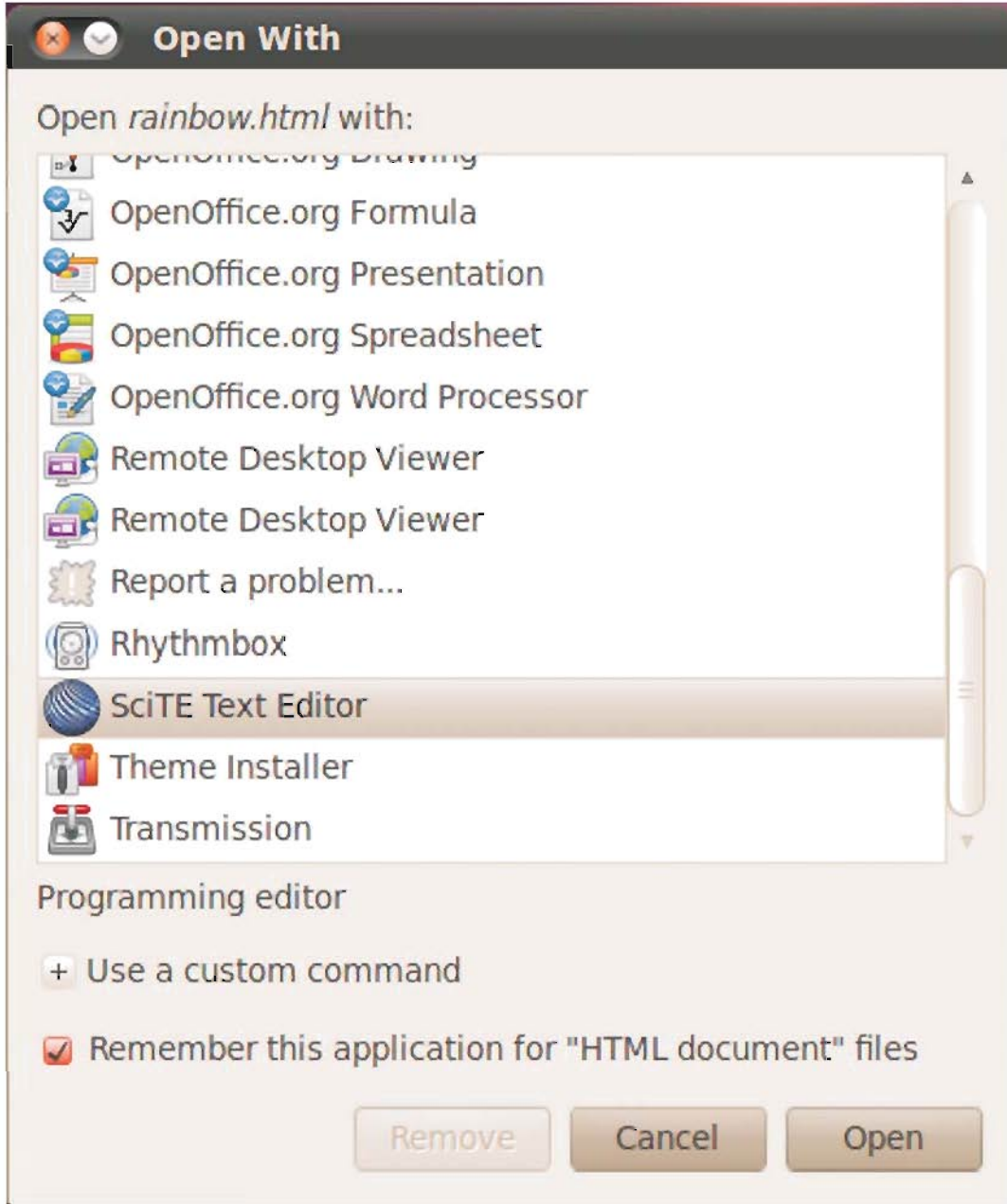
આકૃતિ 1.11 : SciTE એડિટરમાં HTML કોડ

આકૃતિ 1.11માં સ્ક્રીનના ઉપરના ભાગે દર્શાવાયેલ મેનુબાર વિકલ્પો ઉપર ધ્યાન આપો. મેનુ વિકલ્પો ફાઈલ પરની ક્રિયાઓ, લખાણને સુધારવાની સવલતો, માહિતી શોધવા, નિહાળવા તેમજ અન્ય મૂળભૂત સવલતો ધરાવે છે. મેનુ વિકલ્પોની તરત નીચેની લીટીમાં કેટલીક મૂળભૂત ક્રિયાઓ જેવી કે, new, open, save, edit, search વગેરે માટેના આઈકોન રજૂ કરે છે. SciTE એડિટરમાં હયાત કોડને તમે નીચે જણાવેલ પગલાં અનુસરીને ખોલી શકશો.

પગલું 1 : જે ફાઈલને ખોલવી હોય તેને મેળવો.

પગલું 2 : તેના પર જમણું બટન ક્લિક કરો અને Open With પસંદ કરો. આકૃતિ 1.12માં દર્શાવ્યા મુજબનો સ્ક્રીન દેખાશે.

પગલું 3 : SciTE ટેક્સ્ટ એડિટર પસંદ કરો. તે ફાઈલને SciTE એડિટરમાં ખોલશે. SciTE દ્વારા દર્શાવાતા ઇન્ડેન્ટ (indentation) માટેના નિર્દેશો અને ટેગના રંગોની નોંધ લો. આ રીતે ઇન્ડેન્ટ અને ટેગ વિવિધ રંગોમાં હોવાને લીધે લખાણ અને ટેગ જુદા તરી આવે છે અને કોડને વાંચવામાં સરળતા બને છે.



આકૃતિ 1.12 : Open With ડાયલોગ બોક્સ

પગલું 4 : બ્રાઉઝરમાં html કોડનું પરિણામ જોવા માટે Tools → Go પસંદ કરો અથવા F5 કી દબાવો.

HTMLની વર્તમાન આવૃત્તિ

HTMLના વિકાસકાળમાં તેનું ધ્યાન તેના ઉદ્દેશ્ય તરફ કેન્દ્રિત કર્યું, જેથી કરીને લખાણને ટાઈપ કરીને તૈયાર કરવાનું અને પ્રકાશિત કરવાનું અસરકારક રહે. સમયાંતરે HTMLમાં વધુ ને વધુ સવલતો ઉમેરાતી ગઈ જેથી કરીને તે નવા બ્રાઉઝર્સ, નવી ટેકનોલોજી અને સર્જકોની સતત વધતી જતી માંગને અનુરૂપ બની શકે અને ઉપયોગી થઈ શકે. HTMLની અદ્યતન આવૃત્તિને HTML 5.0 તરીકે ઓળખવામાં આવે છે અને તે અત્યારે હાલમાં લોકપ્રિય થઈ રહી છે.

આ પ્રકરણમાં આવરી લીધેલ ટેગ

આ પ્રકરણમાં આપણે કોષ્ટક 1.3માં દર્શાવેલ ટેગની ચર્ચા કરી છે.

ટેગ (Tag)	વર્ણન (Description)
<a href> ... 	મૂળ ફાઈલને લિન્ક દ્વારા જોડવામાં આવેલી (referred) ફાઈલ સાથે સાંકળે છે.
 ..	લખાણને ઘાટા અક્ષરોમાં દર્શાવે છે.
<body>...</body>	HTML દસ્તાવેજનો મુખ્ય ભાગ (Body) નક્કી કરે છે. <html> ટેગની જોડીની વચ્ચે દર્શાવાય છે.
 or 	લીટીનો અંત (Line break) દર્શાવે છે. તે અયુગ્મ ટેગ છે.
<h1>...</h1>	પ્રથમ સ્તરનું મથાળું દર્શાવે છે.
<h2>...</h2>	દ્વિતીય સ્તરનું મથાળું દર્શાવે છે.
<h3>...</h3>	તૃતીય સ્તરનું મથાળું દર્શાવે છે.
<h4>...</h4>	ચોથા સ્તરનું મથાળું દર્શાવે છે.
<h5>...</h5>	પાંચમા સ્તરનું મથાળું દર્શાવે છે.
<h6>...</h6>	છઠ્ઠા સ્તરનું મથાળું દર્શાવે છે.
<head>...</head>	HTML દસ્તાવેજનો મુખ્ય વિભાગ દર્શાવે છે. <html> ટેગની જોડીની વચ્ચે આવે છે.
<html>...</html>	આખા HTML દસ્તાવેજને આવરી લે છે.
<i> ..</i>	લખાણને ત્રાંસા (italics) સ્વરૂપે છાપે છે.
<p>...</p>	ફકરો દર્શાવે છે.
<pre>...</pre>	પૂર્વરૂપાંકિત (preformatted) લખાણ દર્શાવે છે.
<s> ..</s>	છેકી નાંખેલ સ્વરૂપે લખાણ દર્શાવે છે.
<title>...</title>	દસ્તાવેજનું શીર્ષક દર્શાવે છે. <head> નામની ટેગની જોડીની વચ્ચે આવે છે.
<tt> ..</tt>	લખાણને ટાઈપરાઈટર જેવા અક્ષરોમાં દર્શાવે છે.
<u> ..</u>	લખાણને નીચે લીટી દોરેલ હોય તેવા સ્વરૂપે દર્શાવે છે.

કોષ્ટક 1.3 : પ્રકરણ 1માં આવરી લેવાયેલ HTML ટેગ

સારાંશ

આ પ્રકરણમાં આપણે HTMLનો ઉપયોગ કરીને વેબપેઈજ કેવી રીતે તૈયાર કરવાં તે શીખ્યા. HTML એ એવી મૂળભૂત સુવિધા છે, જે વેબની માહિતી કેવી રીતે તૈયાર થાય, પ્રકાશિત થાય અને વેબ બ્રાઉઝરનો ઉપયોગ કરી કેવી રીતે દર્શાવવી તે વર્ણવે છે. આ પ્રકરણમાં, HTML દસ્તાવેજના માળખાના બે મહત્વના ભાગ જેવા કે, HEAD અને BODY સહિત કેટલાક મૂળ ટેગ વર્ણવવામાં આવ્યા. માહિતીનો ઉપયોગ કરીને તમે સાદાં વેબપેઈજ તૈયાર કરી શકો. તૈયાર કરેલાં અનેક વેબપેઈજને તમે આ પ્રકરણમાં ચર્ચા કરેલ એન્કર ટેગ વડે જોડી શકો. HTMLના મૂળભૂત ખ્યાલો અને HTMLનો ઇતિહાસ પૂરો પાડવા ઉપરાંત આ પ્રકરણમાં HTML દસ્તાવેજ તૈયાર કરવા gedit અને SciTE જેવા એડિટર વિશે પણ માહિતી આપવામાં આવી છે.

સ્વાધ્યાય

1. HTMLના ઇતિહાસ અને તેના વિકાસ વિશે ટૂંક નોંધ લખો.
2. HTML દસ્તાવેજનું સરળ ઉદાહરણ આપી HTML દસ્તાવેજના માળખાનું વર્ણન કરો.
3. HTML એલિમેન્ટ એટલે શું ? HTML એલિમેન્ટનું માળખું સમજાવો.
4. HTML દસ્તાવેજને તમારા ડેસ્કટોપ પર કેવી રીતે નિહાળી શકાય ?
5. કોઈ પણ ત્રણ બ્રાઉઝરની યાદી બનાવો.
6. HTML મથાળાની વિવિધ ઢબ ઉદાહરણ સાથે ચર્ચો.
7. HTMLના વિવિધ ફોર્મેટિંગ ટેગ ઉદાહરણ સાથે ચર્ચો.
8. HTMLની પૂર્વરૂપાંકિત (preformatted) ટેગની ઉદાહરણ સાથે ચર્ચા કરો.
9. HTMLના એન્કર ટેગ વિશે ટૂંક નોંધ લખો.
10. HTMLમાં પૂર્ણ અને સંબંધિત સરનામાની વ્યાખ્યા આપી ચર્ચા કરો.
11. નીચેના પ્રશ્નો માટે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :
 - (1) વેબની માહિતી દર્શાવવા માટે કઈ માર્ક-અપ ભાષાની જરૂર પડે છે ?
(a) CML (b) HTML (c) NML (d) WML
 - (2) નીચેનામાંથી વેબપેઈજ વર્ણવવા માટેની ભાષા કઈ ગણાય છે ?
(a) HTML (b) WML (c) NML (d) CML
 - (3) નીચેનામાંથી HTMLનું પૂરું નામ કયું છે ?
(a) Hot Text Manipulation Language (b) Hyper Text Manipulation Law
(c) Hyper Text Markup Language (d) Hidden Text Markup Language
 - (4) નીચેનામાંથી SGMLનું પૂરું નામ કયું છે ?
(a) Standardized General Markup Language
(b) System General Manipulation Law
(c) Standardized Genome Markup Law
(d) Standardized Gigabyte Markup Language
 - (5) નીચેનામાંથી HTML એલિમેન્ટને કોણ સંબોધે છે ?
(a) શરૂઆતની ટેગ, લખાણ અને અંતની ટેગ (b) ખૂણિયા કૌંસ
(c) લખાણ (d) આપેલામાંથી કોઈ પણ
 - (6) HTML એલિમેન્ટની સાથે વધારાની ગોઠવણ કરવા નીચેનામાંથી શું ઉપયોગમાં લેવાય છે ?
(a) સંખ્યા (નંબર) (b) ગુણધર્મો (એટ્રીબ્યુટ)
(c) ટિપ્પણી (કોમેન્ટ) (d) લખાણ (કન્ટેન્ટ)
 - (7) નીચેનામાંથી કયો વિકલ્પ ખાલી ટેગ રજૂ કરે છે, જેમાં કોઈ લખાણની જરૂર પડતી નથી ?
(a) Compete (b) Empty (c) Null (d) Void
 - (8) નીચેનામાંથી કયા ગુણધર્મનો પ્રકાર કોઈ પણ ટેગ સાથે આવી શકે ?
(a) અજોડ (Unique) (b) સાર્વત્રિક (Universal)
(c) બિન અગત્યનો (Trivial) (d) પ્રાથમિક (Preliminary)

(9) HTML દસ્તાવેજમાં કયા પ્રકારની માહિતી સમાવી શકાય ?

- (a) મલ્ટીમીડિયા માહિતી (b) શાબ્દિક માહિતી
(c) ફાઈલનામનું પથનામ અને ફાઈલનું નામ (d) આપેલ તમામ

(10) HTML દસ્તાવેજને સુધારવા માટે નીચેના પૈકી કયું એડિટર છે ?

- (a) SciTE (b) BrTE (c) LigHT (d) SprITE

પ્રાયોગિક સ્વાધ્યાય

1. તમારી શાળા વિશેની પ્રારંભિક માહિતી આપતું વેબપેઈજ બનાવો. આ પાનાનું મથાળું "My School" આપો. જરૂરી ફોર્મેટિંગ અને રજૂઆતકર્તા ટેગનો ઉપયોગ કરો.
2. એક વેબપેઈજ બનાવો, જે તમારા ક્લાસનો પરિચય કરાવે. તેમાં તમારા વર્ગશિક્ષક, અન્ય વિષયના શિક્ષક અને જે વિષયો તમે ભણો છો તેની માહિતીનો સમાવેશ કરો. આ પાનાનું મથાળું "My School" આપો. જરૂરી ફોર્મેટિંગ અને રજૂઆતકર્તા ટેગનો ઉપયોગ કરો.
3. આ સ્વાધ્યાયના પ્રકરણમાં તમે તૈયાર કરેલ વેબપેઈજને બોલાવવા માટે આ સ્વાધ્યાયના પ્રશ્ન 1 માં તૈયાર કરેલ વેબપેઈજને સુધારો. “હોટ ટેક્સ્ટ”ને એ રીતે ગોઠવો કે, જેથી જ્યારે 'My School' પાના પરથી ક્લિક કરવામાં આવે ત્યારે તે 'My Class' પાના પર પહોંચી જાય.





HTMLમાં Head અને Body વિભાગ

Head વિભાગ

કોઈ પણ HTML દસ્તાવેજને Head અને Bodyના નામે ઓળખાતા બે ભાગમાં વહેંચવામાં આવે છે. HTML દસ્તાવેજનો head વિભાગ દસ્તાવેજ વિશેની જરૂરી માહિતી પ્રદાન કરે છે. Head વિભાગના લખાણને <head> અને </head> નામના બે ટેગની વચ્ચે લખવામાં આવે છે. આ બન્ને ટેગ અને તેની વચ્ચેના લખાણને head એલિમેન્ટ તરીકે ઓળખવામાં આવે છે.

Head વિભાગમાં સમાવી શકાતી પ્રથમ જરૂરી બાબત છે, શીર્ષક (title). અહીં એ ખાસ નોંધવું જોઈએ કે આ શીર્ષક વેબપેઈજના લખાણ તરીકે દર્શાવાશે નહીં તે માત્ર વેબપેઈજને દર્શાવતી બ્રાઉઝર વિન્ડોના શીર્ષક તરીકે દર્શાવાય છે.

અહીં એ પણ ખાસ નોંધવું જોઈએ કે, જો head વિભાગમાં શીર્ષક લખવામાં ન આવ્યું હોય તો, મોટાભાગના બ્રાઉઝર દ્વારા બ્રાઉઝર વિન્ડોના શીર્ષક તરીકે URL પણ અથવા ફાઈલનું નામ દર્શાવાય છે. વધુમાં, title ટેગને અંતે પૂરો થતો ટેગ લગાડવાનું ન ભૂલવું જોઈએ. જો અંત ટેગ લગાડવાનું ભૂલી જવાયું હશે તો આખા લખાણને શીર્ષક માની લેવામાં આવશે અને કદાચ એવું પણ બને કે, HTML દસ્તાવેજને બ્રાઉઝર વિન્ડોમાં લાવવામાં ન આવે.

Head વિભાગમાં HTML દસ્તાવેજ અને લખાણ બાબતે કેટલીક વધારાની માહિતી પણ આવે છે. વધારાની માહિતી આપતા ટેગને મેટા-ટેગ (meta-tag) તરીકે ઓળખવામાં આવે છે. મેટા-ટેગનો ઉપયોગ મોટે ભાગે બ્રાઉઝર અને સર્ચ એન્જિન(Search Engine)ને સંબંધિત માહિતી સંગ્રહવા માટે થાય છે. યોગ્ય અને બંધબેસતાં મેટા-ટેગનો ઉમેરો વેબપેઈજની પ્રકૃતિ વર્ણવે છે અને સર્ચ એન્જિન દ્વારા વેબપેઈજને અસરકારક રીતે શોધવાનું સરળ બનાવે છે.

વર્ણન

મોટાભાગના સર્ચ એન્જિન ઉપયોગકર્તાને જ્યારે શોધનું પરિણામ રજૂ કરશે ત્યારે તે વર્ણન દર્શાવશે. આવા વર્ણનની અનુપસ્થિતિના કિસ્સામાં, સર્ચ એન્જિન માત્ર શરૂઆતના થોડાક શબ્દો દર્શાવશે. મેટા-ટેગના description ગુણધર્મનું ઉદાહરણ નીચે મુજબ છે:

<meta name="DESCRIPTION" content="About rainbow and its colours">

ચાવીરૂપ શબ્દો

આ ટેગમાં આપવામાં આવેલ ચાવીરૂપ શબ્દો (Keywords) સર્ચ એન્જિન દ્વારા ઉપયોગમાં લેવાશે. અગત્યની લાક્ષણિકતાઓનાં નામ, વેબપેઈજનો ઉદ્દેશ અને અગત્યના મુદ્દાને કી-વર્ડ તરીકે રજૂ કરી શકાય. Keywords વર્ણવતા મેટા-ટેગ ગુણધર્મ નીચે મુજબ છે:

<meta name="KEYWORDS" content="Rainbow, VIBGYOR">

લેખક

નીચેનો મેટા-ટેગ ગુણધર્મ, વેબપેઈજના સર્જક કે લેખક (Author) વિશેની માહિતી પ્રદાન કરે છે.

<meta name="AUTHOR" content="M K Gandhi">

ટિપ્પણી (Comments)

ટિપ્પણી (કોમેન્ટ્સ) HTML કોડમાં વધારાની માહિતી આપવાની તક આપે છે. કોમેન્ટ્સ પરિણામ સ્વરૂપે દર્શાવવા માટે નથી હોતી. એટલે કે, HTMLના સોર્સ કોડ સિવાય કોઈ પણ જગ્યાએ ટિપ્પણી જોઈ શકાશે નહીં. ટિપ્પણી મૂકવા માટે આપણે '<!--' અને '-->' ટેગનો ઉપયોગ કરવો પડે. '<!--' ટેગ ટિપ્પણીની શરૂઆત દર્શાવે છે જ્યારે '-->' ટેગ દ્વારા ટિપ્પણીનો અંત દર્શાવાય છે. ટિપ્પણીને “આમુખ” (prologue) તરીકે પણ ઓળખવામાં આવે છે. <!-- This is a comment --> એ ટિપ્પણીનું એક ઉદાહરણ છે.

અન્ય મેટા-ટેગ

એવા પણ કેટલાક મેટા-ટેગ છે, જે વેબપેઈજને અમુક સેકન્ડ પછી કોઈ ચોક્કસ URLને લઈ આવવાનું કહે અથવા તેને કહી દે કે પાનાં ન આવવાં જોઈએ. નીચેનું ઉદાહરણ દર પાંચ સેકન્ડ પછી વેબપેઈજને પુનઃ લાવી (reload કરી) વેબપેઈજને રીફ્રેશ (refresh) કરે છે.

<meta http-equiv="REFRESH" content="5">

નીચેનું ઉદાહરણ http://test.com/ નામના URL ના લખાણને દર પાંચ સેકન્ડે પુનઃ તાજું કરશે.

<meta http-equiv="refresh" content="5" URL='http://test.com/'>

HTML base

<base> ટેગ વેબપેઈજની તમામ સંબંધિત URL માટે આધારરૂપ URL/ટાર્ગટ નિશ્ચિત કરે છે.

<head>

<base href="http://test.com/ " >

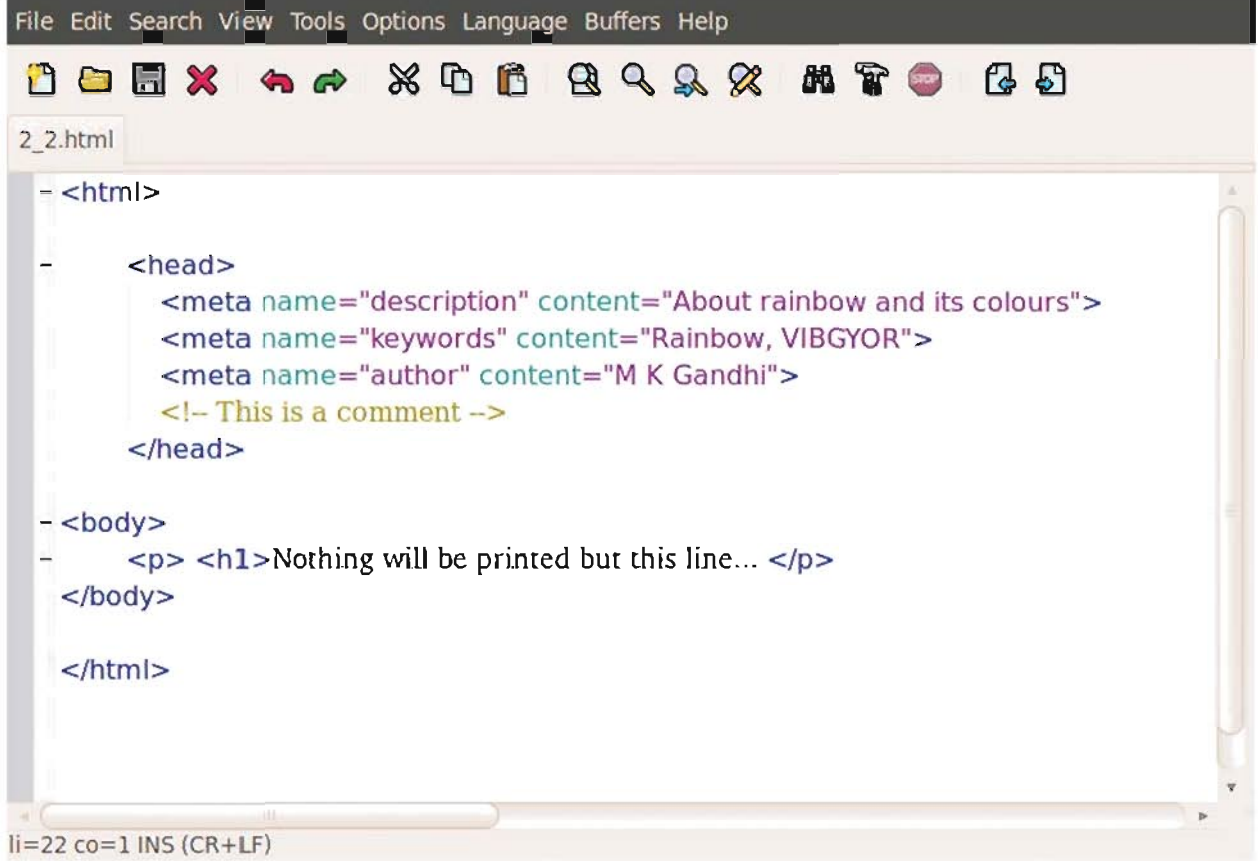
</head>

ટેબલ 2.1માં સારાંશરૂપે કેટલાક અગત્યના HTML head એલિમેન્ટ રજૂ કરેલ છે.

ટેગ	વર્ણન
<head>	દસ્તાવેજ વિશેની માહિતી વ્યાખ્યાયિત કરે છે.
<title>	દસ્તાવેજનું શીર્ષક વ્યાખ્યાયિત કરે છે.
<base>	વેબપેઈજની તમામ લિન્ક માટે એક સામાન્ય સરનામું અથવા ગંતવ્ય સ્થાન વ્યાખ્યાયિત કરે છે.
<link>	દસ્તાવેજ અને બાહ્ય સંસાધન વચ્ચે સંબંધ વ્યાખ્યાયિત કરે છે.
<meta>	HTML દસ્તાવેજ અંગેના મેટા-ડેટા વ્યાખ્યાયિત કરે છે.
<script>	ક્લાયન્ટ તરફની સ્ક્રિપ્ટ વ્યાખ્યાયિત કરે છે.
<style>	દસ્તાવેજ માટેની ઢબની માહિતી વ્યાખ્યાયિત કરે છે.

કોષ્ટક 2.1 : HTML દસ્તાવેજનું Head એલિમેન્ટ

head એલિમેન્ટ સાથે અવલોકન માટે આકૃતિ 2.1માં દર્શાવેલ HTML કોડને ધ્યાને લો.



```
<html>

  <head>
    <meta name="description" content="About rainbow and its colours">
    <meta name="keywords" content="Rainbow, VIBGYOR">
    <meta name="author" content="M K Gandhi">
    <!-- This is a comment -->
  </head>

  <body>
    <p> <h1>Nothing will be printed but this line... </p>
  </body>

</html>
```

આકૃતિ 2.1 : HTML ના Head એલિમેન્ટનો પ્રયોગ

ઉપરના કોડનો અમલ કરો અને બ્રાઉઝરમાં તેનું પરિણામ જુઓ. તે બિલકુલ આકૃતિ 2.2 જેવું દેખાશે.



Nothing will be printed but this line...

Done

આકૃતિ 2.2 : આકૃતિ 2.1માં દર્શાવેલ HTML કોડનું પરિણામ

Body વિભાગ

HTML દસ્તાવેજનો મુખ્ય વિભાગ <body> અને </body> ટેગની વચ્ચે લખવામાં આવે છે. HTMLનું body એલિમેન્ટ બ્રાઉઝરની અંદર જેને દર્શાવવાની હોય તે બાબતોના સંગ્રાહક તરીકે કાર્ય કરે છે.

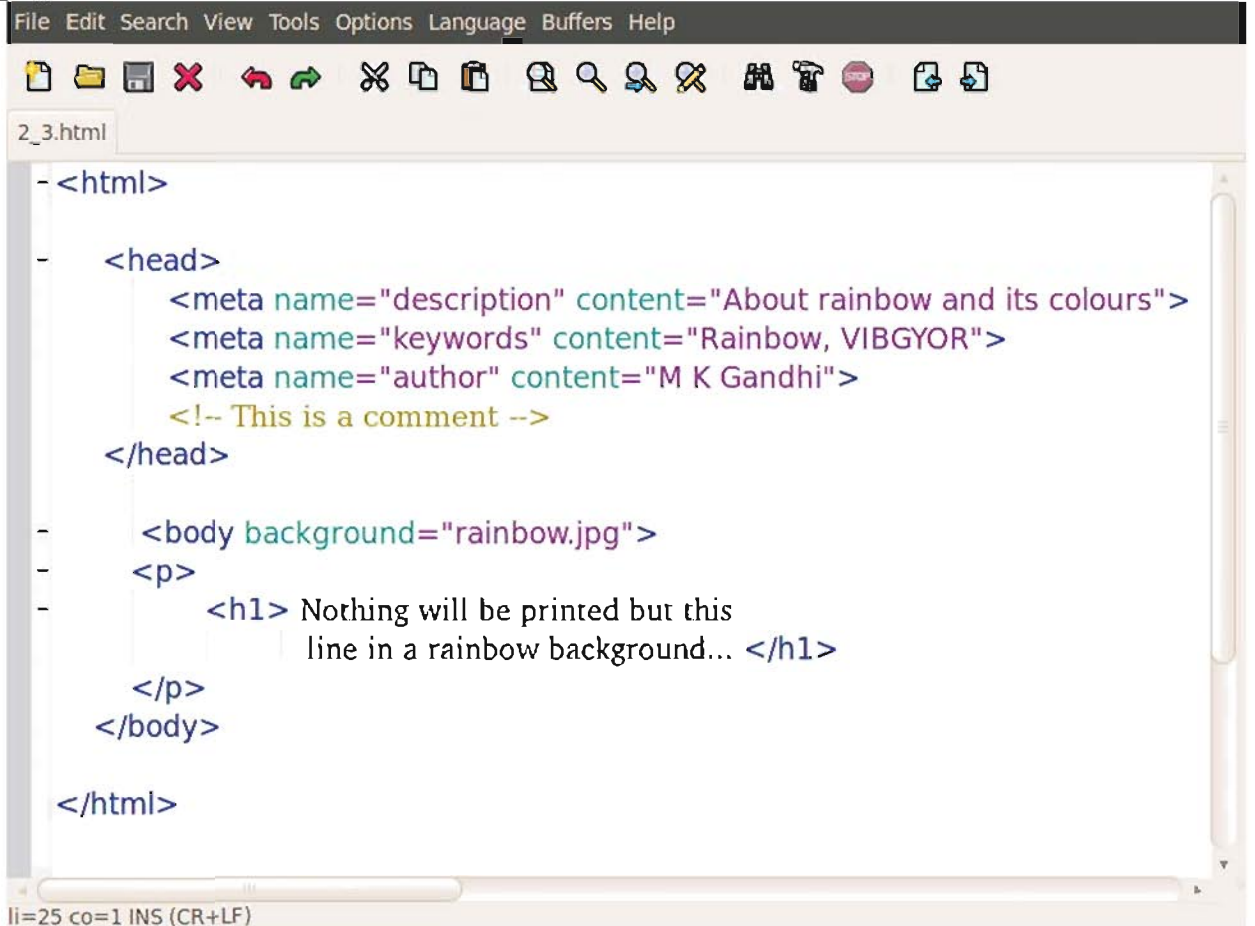
આગળના પ્રકરણમાં આપણે મથાળાં, ફકરા અને લખાણના સ્વરૂપાંકન માટેનાં કેટલાંક ટેગ વિશે વાત કરી કે જે body વિભાગમાં આવી શકે. સામાન્ય રીતે, આ ટેગની વચ્ચે કંઈક લખાણ જરૂરી હોય છે. એવા પણ કેટલાક ટેગ હોય છે જેની વચ્ચે કોઈ લખાણની જરૂર હોતી નથી, આવા ટેગને ખાલી ટેગ (Empty Tag) કહે છે. આવા ખાલી ટેગનું ઉદાહરણ છે
. તો ચાલો, HTML દસ્તાવેજના body વિભાગમાં ઉપયોગમાં લેવાતા કેટલાંક વધુ ટેગ અને ગુણધર્મો જોઈએ.

બેકગ્રાઉન્ડ છબી

અત્યાર સુધી, આપણે એવા HTML કોડ જોયા છે, જે સાદા સફેદ બેકગ્રાઉન્ડ પર લખાણ દર્શાવે. સજાવટસભર અને રંગીન બેકગ્રાઉન્ડ પર લખાણને રજૂ કરવા Body ટેગ સાથે આપણે background ગુણધર્મનો પણ ઉપયોગ કરી શકીએ. ટેક્સ્ટનાં બેકગ્રાઉન્ડ માટે આપણે આપણા કમ્પ્યુટર પર ઉપલબ્ધ કોઈ પણ છબીનો ઉપયોગ કરી શકીએ. HTML પાના માટેના બેકગ્રાઉન્ડ ગોઠવવા આપણે નીચે મુજબની ટેગનો ઉપયોગ કરી શકીએ :

<body background="rainbow.jpg">

અહીં background શબ્દ એ એક ગુણધર્મનું નામ છે. ગુણધર્મને એક ચોક્કસ કિંમત આપવી પડે. આ કિંમત તરીકે આપણે એવી ફાઈલનું નામ આપવું જોઈએ (મોટેભાગે છબીની ફાઈલ), કે જેને આપણે બેકગ્રાઉન્ડ તરીકે દર્શાવવા ઇચ્છીએ છીએ. આપણા કિસ્સામાં તે rainbow.jpg છે. આપણે કોઈ પણ માન્ય છબી સ્વરૂપ જેવાં કે jpg, bmp, png અને tiff આપી શકીએ.



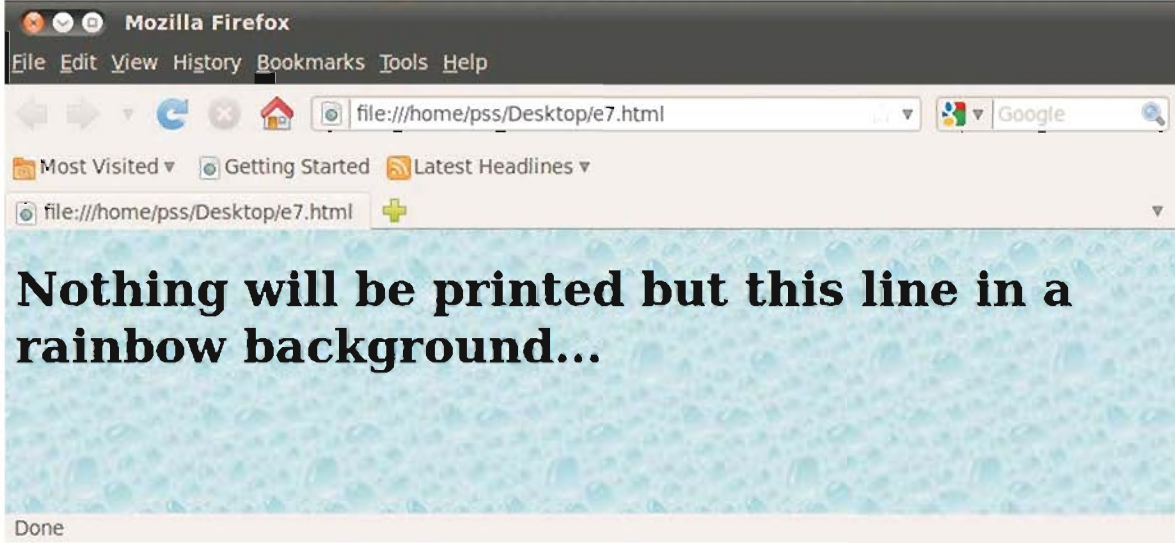
આકૃતિ 2.3 : બેકગ્રાઉન્ડ છબી મૂકવા માટેનો HTML કોડ

તો ચાલો, આકૃતિ 2.3માં દર્શાવ્યા મુજબના HTML કોડ લખીએ કે જે, rainbow.jpg નામની છબી વડે બેકગ્રાઉન્ડ રચે. એ નોંધવું જોઈએ કે, અહીં તમે આ rainbow.jpgને બદલે તમારી પસંદગીની કોઈ પણ છબી ફાઈલને પણ ગોઠવી શકો.

જ્યારે આપણે HTML કોડની અંદર છબી ફાઈલનો ઉપયોગ કરીએ છીએ, ત્યારે આપણે એ ખાતરી કરી લેવી

જોઈએ કે તે ફાઈલનું નામ, તેનું અનુલંબન અને તેનો પથ યોગ્ય રીતે લખવામાં આવ્યો છે કે નહિ. જો ફાઈલનું નામ દર્શાવવા બાબતે કોઈ ભૂલ થશે તો પાનાના બેકગ્રાઉન્ડ તરીકે કંઈ પણ દેખાશે નહીં.

આકૃતિ 2.3માં દર્શાવ્યા મુજબ કોડ ટાઈપ કરો અને તમારી ઈચ્છાનુસાર ફાઈલનાં નામ સાથે સાચવી દો. (અહીં તે e7.html તરીકે સચવાઈ છે.) બ્રાઉઝરમાં કોડને નિહાળો. બેકગ્રાઉન્ડ છબી ઉમેરવાનું પરિણામ આકૃતિ 2.4 દર્શાવે છે.



આકૃતિ 2.4 : બેકગ્રાઉન્ડ તરીકે છબી ઉમેરવી

ઘણી વખત, બેકગ્રાઉન્ડની છબી તેના લખાણ કરતાં વધુ આકર્ષક હોય છે. આપણે એ બાબતની પણ ખાસ કાળજી લેવી જોઈએ કે વાચકનું ધ્યાન લખાણ પરથી હટીને બેકગ્રાઉન્ડ પર કેન્દ્રિત ન થઈ જાય. બેકગ્રાઉન્ડ તરીકે રજૂ કરનાર છબી સુરમ્ય હોવી જોઈએ અને લખાણને વાંચવા પ્રેરે તેવી હોવી જોઈએ નહીં કે, લખાણ ઢંકાઈ જાય તેવી. વળી, જો બેકગ્રાઉન્ડ છબી વધુ પડતી મોટી હશે તો તેને બ્રાઉઝરમાં આવવામાં ઘણો સમય લાગશે. અહીં એ પણ નોંધવું જોઈએ કે, જો બેકગ્રાઉન્ડ માટે ઉપયોગમાં લેવાનારી છબીનું કદ સ્ક્રીનના કદ કરતાં નાનું હશે તો જ્યાં સુધી આખો સ્ક્રીન ભરાઈ નહીં જાય ત્યાં સુધી કદાચ તે છબીની એક પછી એક પ્રતિકૃતિઓ બનતી રહેશે. સામાન્ય રીતે જ્યારે તમે વેબપેઈજને સ્ક્રોલ કરશો ત્યારે બેકગ્રાઉન્ડ છબી સ્ક્રોલ થશે. સિવાય કે, તમે તેને નીચે દર્શાવ્યા મુજબ સ્થિર કરી દો.

<body background="rainbow.gif" bgproperties="fixed">

બેકગ્રાઉન્ડ રંગ

બેકગ્રાઉન્ડમાં છબી દર્શાવવાને બદલે બેકગ્રાઉન્ડ રંગનો ઉપયોગ કરીને પણ આપણે વેબપેઈજને આકર્ષક બનાવી શકીએ. ધારો કે, આપણે આપણા વેબપેઈજનું બેકગ્રાઉન્ડ પીળા રંગનું કરવું હોય તો નીચે મુજબની ટેગથી કામ થઈ જશે.

<body bgcolor="#FFFF00">

અહીં, bgcolor એ બેકગ્રાઉન્ડ રંગ નિર્ધારિત કરતો ગુણધર્મ છે. અહીં એ ખાસ નોંધ લો કે, આપણે નીચે દર્શાવ્યા પ્રમાણે બેકગ્રાઉન્ડ છબી અને રંગ એકસાથે ગોઠવી શકીએ.

<body background="rainbow.jpg" bgcolor="#FFFF00">

આવા કિસ્સામાં, જ્યાં સુધી બ્રાઉઝરમાં સંપૂર્ણપણે છબી આવી ન જાય ત્યાં સુધી બેકગ્રાઉન્ડ રંગ દર્શાવાશે. આવી અસર સામાન્ય રીતે ધીમા કમ્પ્યુટરમાં દેખાશે, ઝડપી કમ્પ્યુટર પર તો કદાચ આપણે કોઈ અસર જોઈ ન શકીએ તેવું બને.

ટેલિવિઝન અને કમ્પ્યુટરના વી.ડી.યુ.ના સ્ક્રીન જેવા ઇલેક્ટ્રોનિક માધ્યમમાં કોઈ પણ રંગ ત્રણ મૂળભૂત રંગ, લાલ (Red), લીલો (Green) અને વાદળી (Blue) નામના ત્રણ રંગોના સંયોજનરૂપે જ ગણાય છે. તેને ટૂંકમાં RGB તરીકે પણ ઓળખવામાં આવે છે.

યાદ કરો, બાળપણમાં તમે બે કે વધુ રંગોનું મિશ્રણ કરી નવો રંગ બનાવતા હતા ! શક્ય છે કે કદાચ જાંબલી રંગ બનાવવા તમે વાદળી અને લાલ રંગનું મિશ્રણ કર્યું હોય !

કમ્પ્યુટરમાં રંગોને સોળઅંકી પદ્ધતિ મુજબ 00 રિડીથી FF રિડી સુધીના સંકેત આપવામાં આવે છે. આમ કોઈ રંગ રજૂ કરવા કેટલોક લાલ, કેટલોક લીલો અને કેટલોક વાદળી (RGB) રંગને રજૂ કરતો છ આંકડાનો કોડ બનાવવો પડે. એટલે કે, દરેક રંગ માટેના બે આંકડા, લાલ રંગ માટેનો કોડ FF0000 થશે. અહીં લાલ રંગને FF કિંમત (દશાંશ સંખ્યા 255ની સમાન સોળઅંકી સંખ્યા), લીલા રંગને 00 અને વાદળી માટેની કિંમત પણ 00 છે. આ ત્રણેય અંકોની મર્યાદા 00થી FF સુધીની છે. (એટલે કે, 0થી 255 સુધી, કુલ 256 કિંમતો) લાલ, લીલા અને વાદળીની 0થી 255 સુધીની કિંમતોનું સંયોજન $(256 \times 256 \times 256)$ 16 કરોડ વિવિધ રંગો આપે છે. સોળઅંકી સ્વરૂપે રંગોની શબ્દમાળા દર્શાવવાથી આપણે વિવિધ રંગ દર્શાવી શકીએ. આકૃતિ 2.5માં રંગોના કેટલાંક ઉદાહરણો આપ્યાં છે.

FFFFFF→	White colour	→	
FF0000→	Red colour	→	
FFFF00→	Yellow colour	→	
000000→	Black colour	→	
FF00FF→	Pink/Magenta colour	→	
0000FF→	Blue colour	→	
00FF00→	Green colour	→	

આકૃતિ 2.5 : રંગોનાં કેટલાંક ઉદાહરણો

આના બદલે તમે નીચે દર્શાવ્યા પ્રમાણે રંગોના નામોનો ઉલ્લેખ પણ કરી શકો:

<body background="rainbow.jpg" bgcolor="Green">

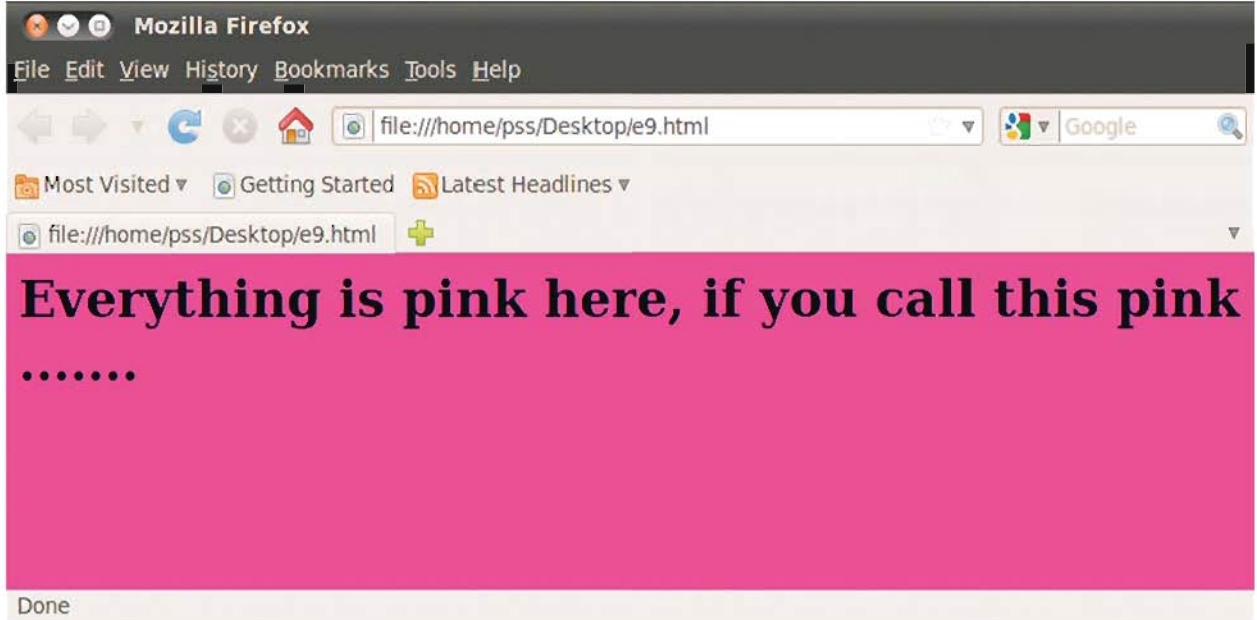
<body background="rainbow.jpg" bgcolor="Chocolate">

રંગોની શબ્દમાળામાં ફેરફાર કરીને તમે અનેકવિધ રંગના બેકગ્રાઉન્ડ તૈયાર કરી શકો. કોડ-લિસ્ટિંગ 2.1માં દર્શાવેલ HTML કોડનો પ્રયત્ન કરી જુઓ.

```
<html>
  <body bgcolor="#FF00FF">
    <h1>Everything is pink here, if you call this pink .....</h1>
  </body>
</html>
```

કોડ-લિસ્ટિંગ 2.1 : બેકગ્રાઉન્ડ ઉમેરવા માટેના HTML કોડ

જ્યારે બ્રાઉઝરમાં નિહાળવામાં આવશે ત્યારે સ્ક્રીન આબેહૂબ આકૃતિ 2.6 જેવો દેખાશે.



આકૃતિ 2.6 : બેકગ્રાઉન્ડ રંગનો ઉપયોગ

લખાણનો રંગ

જેવી રીતે આપણે body ટેગની સાથે ગુણધર્મ(એટ્રીબ્યુટ)ના ઉપયોગ દ્વારા બેકગ્રાઉન્ડ રંગ અને છબીને વ્યાખ્યાયિત કરી તેવી જ રીતે, body ટેગની સાથે text ગુણધર્મ દ્વારા લખાણ(text)ના રંગને વ્યાખ્યાયિત કરી શકીએ. ધારો કે, આપણે નીચે મુજબ એક ઉદાહરણ લખીએ છીએ.

```
<body text="#FF0000" >
```

FF0000 કિંમત વડે text ગુણધર્મનો ઉપયોગ કરવાથી લખાણનો રંગ લાલ થશે. ધારો કે, તમારે ગુલાબી રંગના બેકગ્રાઉન્ડ ઉપર પીળા રંગના અક્ષરોમાં લખાણ દર્શાવવું હોય તો, તમે નીચે મુજબની માત્ર એક સૂચનામાં આમ કરી શકો.

```
<body bgcolor="#FF00FF" text="#FFFF00">
```

લિન્કનો રંગ

દર્શાવાતા વેબપેઈજની અંદર ઘણીવાર તમે કોઈ જુદા સ્થાન / પાન પર નિર્દેશ કરતા વાદળી રંગના શબ્દો જોયા હશે. જ્યારે આવા શબ્દ પર ક્લિક કરવામાં આવે તો તે અન્ય પાન પર લઈ જાય છે. જ્યારે તમે મૂળ પાન પર પાછા આવો ત્યારે તમે એ અવલોકન કર્યું હશે કે આ લખાણનો રંગ બદલાઈ જતો હોય છે. તમે લિન્કની મુલાકાત લીધી હોવાને કારણે વેબપેઈજ દ્વારા આવી લિન્કને મુલાકાત લેવાઈ ગયેલ લિન્ક (visited link) તરીકે ઓળખવામાં આવે છે. આવી મુલાકાત લીધેલ લિન્કને vlink તરીકે ઓળખવામાં આવે છે. જ્યારે જે લિન્કની મુલાકાત લેવાય છે, તે લિન્કને સક્રિય લિન્ક (active link) તરીકે ઓળખવામાં આવે છે. એક્ટિવ લિન્કને alink તરીકે ઓળખવામાં આવે છે. મુલાકાત લેવાઈ ગયેલ લિન્ક અને સક્રિય લિન્કને જુદી પાડવા જુદા જુદા રંગોનો ઉપયોગ કરાય છે. નીચેનું ઉદાહરણ જુઓ :

```
<body alink="#00FF00">
```

```
<body vlink="#FF0000">
```

વેકલ્પિક રીતે તમે આમ પણ લખી શકો.

```
<body vlink="pink">
```

ઉપરના ઉદાહરણમાં ઉપયોગમાં લેવાયેલ alink ગુણધર્મ દસ્તાવેજમાંની સક્રિય લિન્કનો રંગ ગોઠવે છે. સક્રિય લિન્ક જે વખતે ક્લિક કરવામાં આવે તે વખતની લિન્કની પરિસ્થિતિ રજૂ કરે છે. ઉપરના ઉદાહરણમાં ઉપયોગમાં લેવાયેલ vlink ગુણધર્મ દસ્તાવેજની અંદરની હાઈપરલિન્કનો રંગ ગોઠવે છે, જેની મુલાકાત લેવાઈ ગઈ છે.

આડી લીટી (Horizontal line)

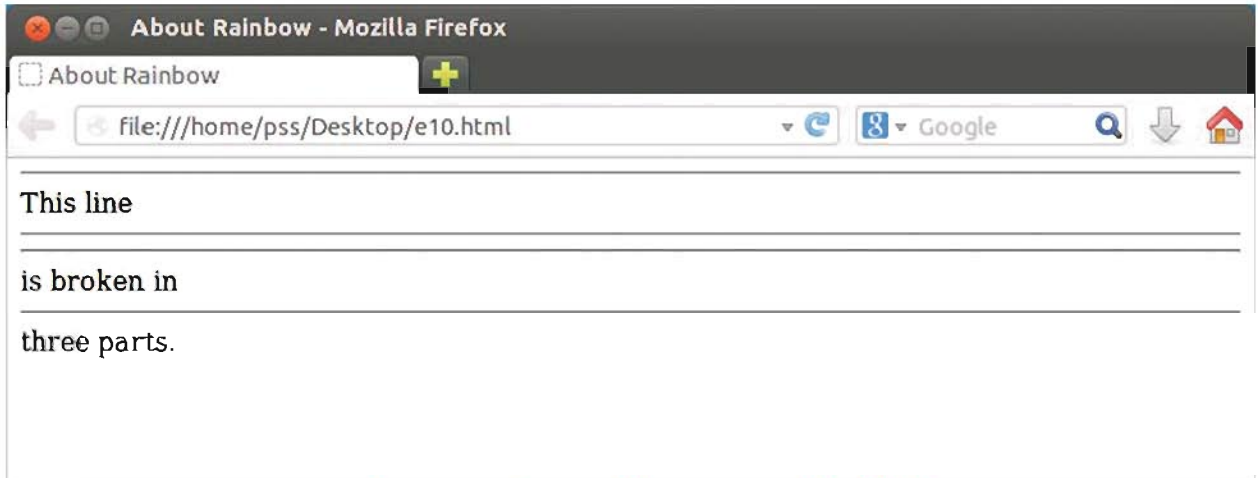
<hr /> એલિમેન્ટ પાનાને સમાંતર આડી લીટી દોરે છે.
 એલિમેન્ટની જેમ જ તે ખાલી એલિમેન્ટ છે. લખાણને બે વિભાગમાં વહેંચીને જુદું પાડવા તેનો ઉપયોગ થાય છે. તેને આડી લીટી પણ કહે છે. આકૃતિ 2.7માં દર્શાવેલ ઉદાહરણ જુઓ:



```
e10.html - SciTE
File Edit Search View Tools Options Language Buffers Help
1 e10.html
- <html>
- <head>
  <title>About Rainbow</title>
</head>
- <body>
  <hr/>
  This line
  <hr/> <hr/>
  is broken in
  <hr/>
  three parts.
</body>
</html>
```

આકૃતિ 2.7 : આડી લીટીનું નિદર્શન

આકૃતિ 2.7માં લખેલ કોડનું પરિણામ આકૃતિ 2.8માં દર્શાવાયું છે.



આકૃતિ 2.8 : બ્રાઉઝર આ રીતે <hr> કમાન્ડને દર્શાવે છે.

આડી લીટી સાથે ઉપયોગમાં લઈ શકાય તેવા પાંચ ગુણધર્મો છે. આ ગુણધર્મો કોષ્ટક 2.2માં આપેલા છે.

ગુણધર્મ	વર્ણન
size	લીટીની જાડાઈ નક્કી કરે છે. તમે 3, 5 કે એથીય વધુ 10નું કદ ધરાવતા પિક્સેલ (pixel)નો ઉપયોગ કરી શકો.
color	Mozilla Firefox જેવા એક્સપ્લોરર (Explorer)માં લીટીનો રંગ નક્કી કરે છે. લીટીનો રંગ સોળ અંકી પદ્ધતિમાં પણ વ્યાખ્યાયિત કરી શકાય છે.
width	લીટીની લંબાઈ નક્કી કરે છે. આ લંબાઈ બ્રાઉઝરની વિન્ડોની પહોળાઈની ટકાવારી તરીકે પણ નક્કી કરી શકાય છે. પહોળાઈની કિંમત સામાન્ય રીતે 100% હોય છે.
align	લીટી જમણી બાજુ, ડાબી બાજુ કે મધ્યમાં ગોઠવાય તે નક્કી કરે છે.
noshade	છાયાંકિત લીટીને બદલે નક્કર સાદી લીટી દર્શાવે છે.

કોષ્ટક 2.2 : આડી લીટીના ગુણધર્મો

ધારો કે, ઉપયોગકર્તા એવું ઇચ્છતો હોય કે 2ના કદની લીટી દર્શાવાય. જે સ્ક્રીનના કદના 25% ભાગ જ ઉપયોગમાં લે અને તે સ્ક્રીનની જમણી બાજુ ગોઠવાયેલી હોય તો નીચે મુજબ ટેગ લખવો પડે.

<hr size=2 width="25%" align="right" >

અહીં એ નોંધવું જરૂરી છે કે HTML માં <hr> ટેગ માટે બંધ થતો ટેગ લખવો ફરજિયાત નથી. આપણે આડી લીટીને કોઈ છબી વડે પણ ભરી શકીએ. તેને માન્ય HTML કોડમાં સમાવીને બ્રાઉઝર દ્વારા નિહાળી શકીએ.

પૂર્વસ્વરૂપાંકિત લખાણનું ઉદાહરણ (Example of preformatted text)

એવી પરિસ્થિતિની કલ્પના કરો કે, જેમાં આપણે આવનારા કોઈ પ્રસંગ માટેની જાહેરાત દર્શાવવા ઇચ્છીએ છીએ. જાહેરાતનો મુસદ્દો (notice) કોષ્ટક 2.3માં દર્શાવ્યા મુજબનો છે.

Practicing Rainbow Colours

Attention Please !

The rainbow is made up of seven colours. When we mix all seven colours they will become a white colour. To demonstrate this practically, we have arranged a laboratory session. The detail of the laboratory session is as follows:

Date : 6 August
Time : 9 : 30 am
Place : First Floor, Lab-1
Instructor : I M Patel

Principal

કોષ્ટક 2.3 : જાહેરાતનો મુસદ્દો

તો ચાલો, સૌપ્રથમ કોષ્ટક 2.3 સ્વરૂપે આપણને આપવામાં આવેલ માહિતી ઉપર થોડું અવલોકન કરીએ.

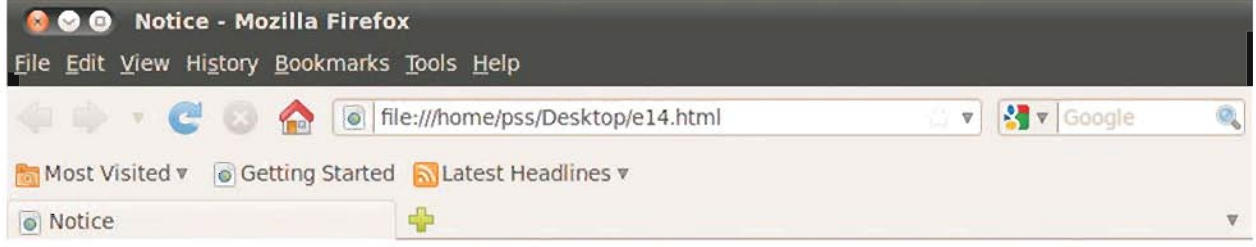
1. કોષ્ટક 2.3માં દર્શાવાયેલ લખાણ HTML દસ્તાવેજના body વિભાગમાં સમાવવું પડે.
2. પ્રથમ લીટી ('Practicing Rainbow Colours') એ મધ્યમાં છે અને મધ્યમાં દર્શાવાયેલ છે. આને યોગ્ય રીતે દર્શાવવા આપણે align="center" ગુણધર્મ સાથે <h1> ટેગનો ઉપયોગ કરવો પડે.
3. બીજી લીટી ('Attention Please !') એ સાદું લખાણ છે. પરંતુ તે નવી લીટી પર મધ્યમાં દર્શાવાયેલ છે તેથી આને એ રીતે દર્શાવવા આપણે align="center" ગુણધર્મ સાથે <p> ટેગ વાપરીશું.
4. એ પછી પ્રયોગશાળામાં થનારા પ્રયોગ વિશે માહિતી આપતું સામાન્ય લખાણ છે. આ લખાણને આપણે સાદા paragraph ટેગ દ્વારા દર્શાવીશું.
5. તારીખ, સમય, સ્થળ અને શિક્ષક વિશેની માહિતી આપતી પછીની ચાર લીટીઓ જુદી રીતે રજૂ કરવામાં આવી છે. આ પૂર્વસ્વરૂપાંકિત લખાણ (preformatted text)નું ઉદાહરણ છે. <pre> અને </pre> ટેગનો ઉપયોગ કરીને આપણે તેને બિલકુલ તેવી જ રીતે દર્શાવી શકીએ.

કોષ્ટક 2.3માં દર્શાવેલ લખાણને રજૂ કરતા HTML કોડ કોડ-લિસ્ટિંગ 2.2માં આપવામાં આવેલ છે.

```
<html>
<head>
<title> Notice </title>
</head>
<body>
<h1 align= "center "> Practicing Rainbow Colours </h1>
<p align =center> Attention Please! </p>
<p> The rainbow is made up of seven colours. When we mix all seven colours
they will become a white colour. To demonstrate the experiment,
we have arranged a laboratory session. The detail of the laboratory
session is as follows :
</p>
<pre>
Date      : 6 August
Time      : 9 : 30 am
Place     : First Floor, Lab-1
Instructor : I M Patel
                Principal
</pre>
</body>
</html>
```

કોડ-લિસ્ટિંગ 2.2 : જાહેરાત દર્શાવવા માટેના HTML કોડ

કોડ દિસ્ટિંગ 2.2ની સૂચનાઓનું પરિણામ આકૃતિ 2.9માં દર્શાવાયું છે.



Practicing Rainbow Colours

Attention Please!

The rainbow is made up of seven colours. When we mix all seven colours they will become a white colour. To demonstrate the experiment, we have arranged a laboratory session. The detail of the laboratory session is as follows:

Date: 6 August
Time: 9:30 am
Place: First Floor, Lab-1
Instructor: I M Patel

Principal

Done

આકૃતિ 2.9 : HTMLમાં પૂર્વસ્વરૂપાંકિત (Preformatted) લખાણ

અક્ષરોનું સ્વરૂપાંકન (Formatting Characters)

લખાણને કેવી રીતે ઘાટા સ્વરૂપે (bold), ત્રાંસા અક્ષર સ્વરૂપે (italics) અને લખાણની નીચે લીટી સ્વરૂપે (underline) દર્શાવવા તે આપણે અગાઉના પ્રકરણમાં જોયું. જ્યારે આપણે લખાણને કેવા સ્વરૂપે રજૂ કરવું તે માટેની ઈચ્છિત ઢબ html કોડ સ્વરૂપે આપીશું તો બ્રાઉઝર આ સૂચનાઓને ચુસ્તપણે અનુસરશે અને લખાણને કહ્યા પ્રમાણે રજૂ કરશે. આવા ટેગ ભૌતિક ઢબ (physical style) ટેગ તરીકે ઓળખાય છે. કોષ્ટક 2.4માં ભૌતિક ઢબ માટેના ટેગ રજૂ કરવામાં આવ્યા છે.

ટેગ	વર્ણન
 ... 	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણને ઘાટા સ્વરૂપે દર્શાવે છે.
<i> ... </i>	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણને ત્રાંસા અક્ષર સ્વરૂપે દર્શાવે છે.
 ... 	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણ જાણે કે લખ્યા પછી છેકી નાંખ્યું હોય તેવા સ્વરૂપે દર્શાવવા લખાણની ઉપર લીટી દર્શાવશે.
_{...}	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણને તેની આગળના લખાણ કરતાં નાના અક્ષરોમાં અને થોડું નીચે દર્શાવે છે.
^{...}	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણને તેની આગળના લખાણ કરતાં નાના અક્ષરોમાં અને થોડું ઉપર દર્શાવે છે.
<tt> ... </tt>	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણને ટાઈપરાઈટર મશીન જેવા ચોક્કસ અક્ષરો સ્વરૂપે દર્શાવે છે.
<u> ... </u>	આ ટેગની વચ્ચે ટાઈપ કરેલ લખાણની નીચે લીટી દોરાયેલ સ્વરૂપે દર્શાવે છે.

કોષ્ટક 2.4 : ભૌતિક ઢબ માટેના ટેગ (Physical Style Tag)

લખાણના સ્વરૂપાંકન માટે અન્ય રસ્તા પણ છે. સ્વરૂપાંકન માટે બ્રાઉઝરને ભૌતિક સૂચનાઓ આપવાને બદલે આપણે બ્રાઉઝરને માત્ર કહેવાનું હોય છે કે, આપણે શું જોઈએ છે ? આ રીતનું સ્વરૂપાંકન કરી આપતી ટેગને તાર્કિક સ્ટાઈલ માટેના ટેગ (logical style tag) કહે છે. નીચે આપેલા ટેગ એ કેટલાંક તાર્કિક સ્ટાઈલના ટેગનાં ઉદાહરણ છે.

** અને **

આ ટેગની જોડી દ્વારા લખાણને ભારપૂર્વક દર્શાવવામાં આવે છે. મહત્વની બાબત જેવી કે “કરવાં જ પડનારાં કાર્યો”(Must be done) અને “અગત્યનું” (Important) આ ટેગનો ઉપયોગ કરીને લખી શકાય. આ ટેગનું ઉદાહરણ નીચે મુજબ છે:

What a beautiful rainbow...!

** અને **

ટેગની આ જોડી દ્વારા લખાણને વધુ ભારપૂર્વક રજૂ કરવામાં આવશે. આ ટેગનું ઉદાહરણ નીચે મુજબ છે.

** What a beautiful rainbow...!**

ઉપરના ઉદાહરણ દ્વારા અને દ્વારા દર્શાવાયેલ લખાણ વચ્ચેનો તફાવત જુઓ. આવા તાર્કિક સ્વરૂપાંકન કરતા ટેગ કોષ્ટક 2.5માં આપવામાં આવ્યા છે.

ટેગ	વર્ણન
<dfn>	લખાણને નિર્ધારિત ઢબે રજૂ કરે છે.
	કોઈ ચોક્કસ લખાણ પર ભાર મૂકવા માટે
<cite>	પુસ્તકનું શીર્ષક, ચલચિત્ર વગેરે જેવા મહત્વના લખાણને રજૂ કરવા.
<code>	કમ્પ્યુટર પ્રોગ્રામિંગ કોડ દર્શાવવા આ લખાણ ચોક્કસ કદના ફોન્ટમાં દર્શાવાશે.
	આ ટેગ લખાણને વધુ ભારપૂર્વક દર્શાવે છે. લખાણ વધુ ઘાટા અક્ષર સ્વરૂપે દર્શાવાય છે.

કોષ્ટક 2.5 : તાર્કિક સ્વરૂપાંકન ટેગ (Logical Style tag)

Font ટેગ

અક્ષરોના ચોક્કસ જાતના મરોડ અને કદ ગોઠવવા font ટેગનો ઉપયોગ કરવામાં આવે છે. કેટલાંક ઉદાહરણ નીચે આપવામાં આવેલ છે.

<p>This is some text!</p>

<p>This is some text!</p>

<p>This is some text!</p>

અહીં તમે એ નોંધ્યું હશે કે, આપણે સોળઅંકી સંખ્યાને બદલે green, red, blue એમ રંગના સીધા નામનો ઉપયોગ કરેલ છે. ઉપરના ઉદાહરણમાં font ટેગની સાથે ગુણધર્મોનો પણ ઉપયોગ કર્યો છે. આ ગુણધર્મો છે size અને color. જેમાં size એ લખાણમાં અક્ષરોનું કદ નક્કી કરે છે અને color એ લખાણ કયા રંગના અક્ષરોમાં દર્શાવાય તે નક્કી કરે છે.

HTML કોડની શરૂઆતમાં જ આપ્યા વેબપેઈજ માટે font અને તેના વિવિધ ગુણધર્મો નક્કી કરી દેવાનું શક્ય છે. આવું કરવા નીચે મુજબ લખવું પડે:

<basefont face="Arial" size="16">

કોડ લિસ્ટિંગ 2.3માં દર્શાવ્યા પ્રમાણે પૂરા HTML કોડ લખીને ઉપરોક્ત કોડને ચકાસી શકાય.

```
<html>

<body>

    <basefont face = Arial size=16>

    <p><font color="red">This is some text!</font></p>

    <p><font color="blue">This is some text!</font></p>

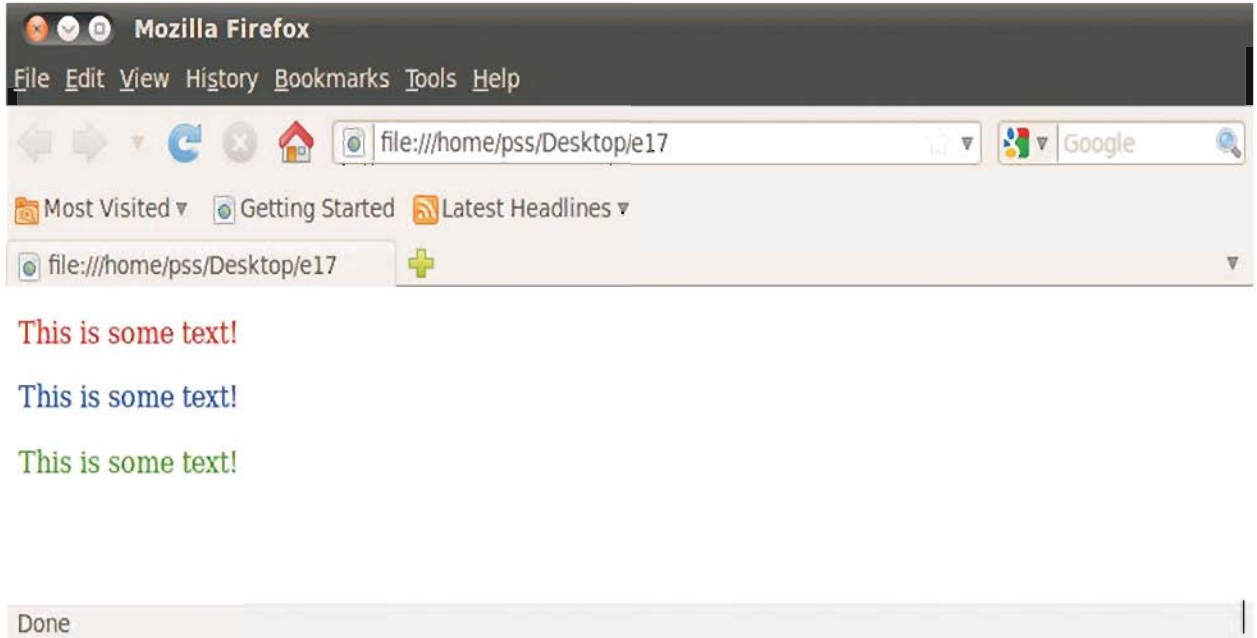
    <p><font color="green">This is some text!</font></p>

</body>

</html>
```

કોડ લિસ્ટિંગ 2.3 : font ટેગને ચકાસવા માટે HTML કોડનો નમૂનો

ઉપરોક્ત કોડને ટાઈપ કરી બ્રાઉઝર દ્વારા નિહાળવાનો પ્રયત્ન કરો. આ HTML કોડનું પરિણામ આબેહૂબ આકૃતિ 2.10માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 2.10 : આધારરૂપ ફોન્ટ અને રંગ

નોંધ :

ધ્યાનથી જુઓ આકૃતિ 2.10માં દેખાતું ફાઈલનું નામ e17.htmlને બદલે e17 છે. સામાન્ય રીતે લિનક્સ ફાઈલના અનુલંબનનો ઉપયોગ કરતું નથી. જ્યારે ફાઈલના નામની પાછળ અનુલંબનનો ઉપયોગ કરવામાં આવે છે ત્યારે તે ફાઈલની અંદરના લખાણ અથવા તેનો ઉપયોગ દર્શાવે છે. ઉદાહરણ તરીકે આપણા કિસ્સામાં HTML અનુલંબન એવું કહે છે કે આ HTML પ્રકારની ફાઈલ છે. ઈચ્છિત પરિણામ મેળવવા લિનક્સ પર ઉપયોગકર્તાએ ફાઈલનું અનુલંબન દર્શાવવાની જરૂર રહેતી નથી. માત્ર જે તે ફાઈલને ખોલવા યોગ્ય પ્રોગ્રામનો ઉપયોગ કરાયો છે કે કેમ તેની ખાતરી કરવાની રહે છે.

ખાસ ચિહ્નો

ખાસ ચિહ્નો જેવાં કે, <, &, ©, વગેરે ચિહ્ન પૂર્વગ તરીકે & (એમ્પરસન્ડ ચિહ્ન) સાથે વેબપેઈજ પર સમાવી શકાય છે. આવી પ્રતીકાત્મક સંજ્ઞાને બદલે આપણે જે તે ચિહ્ન માટેની ASCII કિંમત પણ રાખી શકીએ. આની જરૂર એટલા માટે પડે છે કેમ કે, HTML દસ્તાવેજમાં સીધેસીધા '<' અને '>' ચિહ્નો ઉપયોગ તો કરી ન શકાય. કારણ કે, તે તો ટેગ સાથે ભળી જવાની શક્યતા છે. કોષ્ટક 2.6માં દર્શાવેલ ખાસ ચિહ્નોનો ઉપયોગ કરો.

ચિહ્ન	વર્ણન	સાંકેતિક સંજ્ઞા	કિંમત
<	ના કરતાં નાનું (Less than)	lt	60
>	ના કરતાં મોટું (Greater than)	gt	62
&	એમ્પરસન્ડ (Ampersand)	amp	38
©	કૌપીરાઈટ (Copy right)	–	169
¼	એક ચતુર્થાંશ (One quarter)	–	188
½	અડધું (Half)	–	189
¾	ત્રણ ચતુર્થાંશ (Three quarters)	–	190
®	રજિસ્ટર્ડ ટ્રેડમાર્ક (Registered trademark)	erg	174

કોષ્ટક 2.6 : ખાસ ચિહ્ન માટે સાંકેતિક સંજ્ઞા અને ASCII કિંમત

HTML કોડમાં, ના કરતાં ઓછું (<) અને ના કરતાં વધુ (>) ચિહ્નના ઉપયોગને રજૂ કરતું ઉદાહરણ...

<p align="center"> < Basic Information ></p>

ઉપરોક્ત કોડ વેબપેઈજ પર નીચે મુજબ લખાણ દર્શાવશે:

< Basic Information >

આ પ્રકરણમાં આવરી લેવાયેલ ટેગ

આ પ્રકરણમાં આપણે કોષ્ટક 2.7માં દર્શાવેલ ટેગની ચર્ચા કરી છે.

ટેગ	વર્ણન
<!-- ...-->	HTML કોડમાં ટિપ્પણીના લખાણને ઉમેરવા
	લખાણને ઘાટા અક્ષરોમાં દર્શાવે છે.
<base>	વેબપેઈજમાં તમામ સંબંધિત URL માટે આધારરૂપ URL / ગંતવ્ય દર્શાવે છે.
<basefont>	આખા વેબપેઈજ માટેના સામાન્ય (default) ફોન્ટ વ્યાખ્યાયિત કરે છે.
<body>	HTML દસ્તાવેજ માટે body વિભાગ વ્યાખ્યાયિત કરે છે. તેની સાથે background image, background colour, text colour, link colour વગેરે ગુણધર્મ ઉપલબ્ધ બને છે.

ટેગ	વર્ણન
<cite>	પુસ્તકનું શીર્ષક, ફિલ્મનું નામ વગેરે જેવા મહત્વના શબ્દોને રજૂ કરવા આવું લખાણ ત્રાંસા અક્ષર (italics) સ્વરૂપે દર્શાવાય છે.
<code>	કમ્પ્યુટર પ્રોગ્રામિંગ કોડના વિભાગને રજૂ કરે છે. આવું લખાણ નિયત કદના ફોન્ટમાં દર્શાવાય છે.
<dfn>	લખાણને નિર્ધારિત સ્ટાઇલે રજૂ કરવા આવું લખાણ ત્રાંસા અક્ષર (italics) સ્વરૂપે દર્શાવાય છે.
	લખાણ ભારપૂર્વક રજૂ કરાય છે.
	ચોક્કસ પ્રકારના અને કદના ફોન્ટ ગોઠવવા ફોન્ટ ટેગ ઉપયોગમાં લેવાય છે.
<head>	HTML દસ્તાવેજનો Head વિભાગ દર્શાવવા.
<hr>	આડી લીટી વ્યાખ્યાયિત કરે છે. તેની સાથે size, colour, width, alignment, noshade વગેરે જેવા ગુણધર્મો હોય છે.
<i>	લખાણને ત્રાંસા અક્ષર સ્વરૂપે રજૂ કરે છે.
<link>	દસ્તાવેજ અને બાહ્ય સંસાધન વચ્ચે સંબંધ વ્યાખ્યાયિત કરે છે.
<meta>	સર્ચ એન્જિન અને અન્ય યુટીલીટી પ્રોગ્રામને author, keywords, description, purpose વગેરે વિશે વધારાની માહિતી પૂરી પાડે છે.
<script>	ક્લાયન્ટ બાજુની સ્ક્રિપ્ટ વ્યાખ્યાયિત કરે છે.
<strike>	લખાણને જાણે કે લખ્યા પછી છેકી નાંખ્યું હોય તેવી રીતે દર્શાવે છે.
	લખાણને વધુ ભારપૂર્વક દર્શાવવામાં આવે છે.
<style>	દસ્તાવેજ માટે સ્ટાઇલ અંગેની માહિતી વ્યાખ્યાયિત કરે છે.
<sub>	આ ટેગની વચ્ચે ટાઇપ કરેલ લખાણને તેની આગળના લખાણ કરતાં નાના અક્ષરોમાં અને થોડું નીચે દર્શાવે છે.
<sup>	આ ટેગની વચ્ચે ટાઇપ કરેલ લખાણને તેની આગળના લખાણ કરતાં નાના અક્ષરોમાં અને થોડું ઉપર દર્શાવે છે.
<tt>	આ ટેગની વચ્ચે ટાઇપ કરેલ લખાણને ટાઇપરાઈટર મશીન જેવા ચોક્કસ અક્ષરો સ્વરૂપે દર્શાવે છે.
<u>	આ ટેગની વચ્ચે ટાઇપ કરેલ લખાણની નીચે લીટી દોરાયેલ સ્વરૂપે દર્શાવે છે.

કોષ્ટક 2.7 : પ્રકરણ 2માં ચર્ચેલ HTML ટેગ

એક ઉદાહરણ

તો ચાલો, રોજિંદા જીવનનું એક ઉદાહરણ લઈએ. ધારો કે, તમારે તમારી શાળાની એક સાદી વેબસાઈટ બનાવવી છે. વેબસાઈટ શાળાના સંસાધનો અને પ્રવૃત્તિઓની રજૂઆત કરતી હોય. શાળાની વેબસાઈટનું મુખ્ય પાનું અનુક્રમણિકા (index) અથવા હોમપેજ તરીકે ઓળખી શકાય. પ્રથમ વેબપેઈજ પર શાળાનું નામ પ્રકાશિત થશે. આ હોમપેજ પર શાળાનું ટ્રસ્ટ સાથેનું જોડાણ, નોંધણી નંબર, સંપૂર્ણ સરનામું અને સંપર્કસૂત્રની વધારાની માહિતી પણ મૂકી શકાય. તદ્ઉપરાંત હોમપેજને અન્ય વિવિધ વેબપેઈજ જેવાં કે, 'શૈક્ષણિક પ્રવૃત્તિ', 'ફોટો ગેલેરી', 'કાર્યક્રમો અને પ્રસંગો' વગેરે સાથે જોડાણ (લિન્ક) હોઈ શકે.

આ વેબપેઇજનું વર્ણન નીચે મુજબ આપેલ છે:

- મારી શાળાનું હોમ પેજ : શાળાની વેબસાઇટના મુખ્ય પાના પર અન્ય પાનાના મેનૂ ઉપરાંત શાળાનું નામ, શાળાના ટ્રસ્ટની ટૂંકમાં માહિતી, શાળાનો નોંધણી નંબર, કોઈ સરકારી વિભાગ સાથે શાળાનું જોડાણ અને શાળાના સંપર્ક માટેની જરૂરી વિગતોનો સમાવેશ થશે.
- શિક્ષકગણ અને શૈક્ષણિક પ્રવૃત્તિ : આ પાના પર શિક્ષકગણનો ટૂંકો પરિચય રજૂ થશે. જેમાં દરેક શિક્ષકનું નામ, શૈક્ષણિક લાયકાત, કયા વર્ગમાં તેઓ શીખવે છે તે, અને તેમનું વૈશિષ્ટીકરણ તેમજ તેમનું ઇ-મેઇલ સરનામું હશે.
- ફોટો ગેલરી : આ પાનું વર્ગના ફોટા, ગ્રંથાલય, પ્રયોગશાળા અને તાજેતરના પ્રસંગો દર્શાવે છે. જ્યારે છબીઓનું સંચાલન કરવાનું શીખી જાઓ ત્યારબાદ પછીથી આ વેબપેઇજ બનાવી શકો છો.
- પરિપત્રો અને સૂચનાપત્રો : આ પાનું હવે પછી યોજાનારા કાર્યક્રમો અને પ્રસંગો વિશેની માહિતી ધરાવે છે. તેમાં વિદ્યાર્થીઓ અને વાલીઓ માટેના પરિપત્રો અને સૂચનાપત્રો પણ મૂકાશે. ઉદાહરણ તરીકે, રમતોત્સવ, નિબંધસ્પર્ધા અંગેની સૂચના, શિષ્યવૃત્તિ કાર્યક્રમની જાહેરાત અને પરિણામ.

કોડ-લિસ્ટિંગ 2.4માં આપેલ HTML કોડ જુઓ.

```
<html>
<head>
  <title>About my school</title>
</head>
<body vlink="purple" >
<h1> My School  </h1>

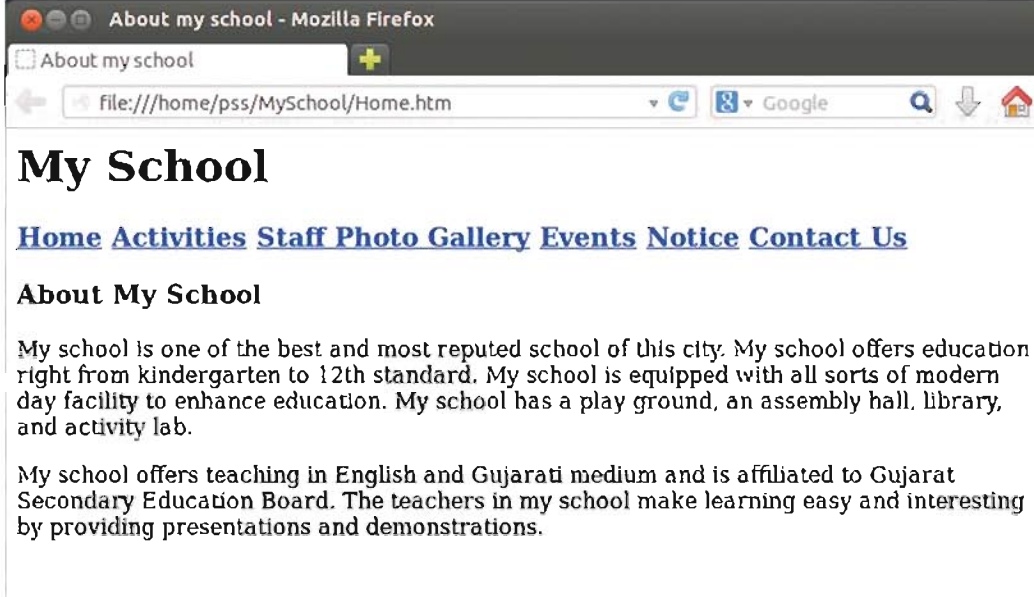
<p> <h3>
  <a href="Home.html">Home</a>
  <a href="Activities.html">Activities</a>
  <a href="Staff.html"> Staff </a>
  <a href="Gallery.html">Photo Gallery</a>
  <a href="Event.html">Events</a>
  <a href="Notice.html">Notice</a>
  <a href="Contact.html">Contact Us</a>
</h3>
</p>
<h3>About My School</h3>

<p> My school is one of the best and most reputed school of this city. My school
offers education right from kindergarten to 12th standard. My school is equipped
with all sorts of modern day facility to enhance education. My school has a
play ground, an assembly hall, library, and activity lab. </h3>
</p>

<p> My school offers teaching in English and Gujarati medium and is affiliated to
Gujarat Secondary Education Board. The teachers in my school make learning
easy and interesting by providing presentations and demonstrations. </h3>
</p>
</body>
</html>
```

કોડ-લિસ્ટિંગ 2.4 : My School વેબસાઇટના મુખ્ય પાના માટેના HTML કોડ

એડિટરનો ઉપયોગ કરી કોડ-લિસ્ટિંગ 2.4માં દર્શાવેલ HTML કોડ ટાઇપ કરો. તેને Home.html તરીકે સાચવો. આ કોડને બ્રાઉઝરમાં નિહાળો. એનું પરિણામ આબેહૂબ આકૃતિ 2.11માં દર્શાવ્યા મુજબનું દેખાશે.



આકૃતિ 2.11 : My School વેબસાઈટનું હોમપેજ

જો તમે Home સિવાયની કોઈ પણ લિન્ક પર ક્લિક કરશો તો ફાઈલ મળતી ન હોવા અંગેના ભૂલ સંદેશ દર્શાવાશે. આ ભૂલસંદેશ અત્યારે એટલા માટે દર્શાવાય છે, કારણ કે આપણે Home.html સિવાય અન્ય કોઈ ફાઈલ તૈયાર કરેલ નથી.

આવી ફાઈલો તૈયાર કરવા માટે નીચે આપેલાં પગલાંને અનુસરો :

- અન્ય ફાઈલ જેવી કે, activities, staff, photo gallery, notice અને contact માટેના HTML કોડ લખો. તેમને યોગ્ય નામ આપી તમે જ્યાં home.html ફાઈલને સાચવેલ છે તે જ ડિરેક્ટરીમાં બધાનો સંગ્રહ કરો. photo gallery માટેનું વેબપેઈજ બનાવવાનું તમે અત્યારે છોડી દઈ શકો છો. હવે પછીના પ્રકરણમાં જ્યારે તમે HTML દસ્તાવેજમાં છબીઓને સમાવવાનું શીખી જાવ એ પછી આ ઉદાહરણરૂપ પ્રોજેક્ટને સુધારવા પ્રયત્ન કરજો. એ પછી તમે તમારી શાળાનું પ્રતીક (logo) પણ તમે ઉમેરી શકો. આ તબક્કે તમે માત્ર વેબપેઈજના બેકગ્રાઉન્ડ તરીકે કોઈ ચિત્ર કે છબી મૂકવા સમર્થ છો.
- "Practicing Rainbow Colours" વિશે કોડ લિસ્ટિંગ 2.2માં આપેલ માર્ગદર્શન અનુસાર તમે એક સૂચનાપત્ર (notice) તો તૈયાર કરી જ દીધેલ છે. તમે આ ફાઈલનું નામ બદલીને Notice.html તરીકે સાચવી શકો.
- ફોર્મેટિંગ ટેગ ઉપરાંત જરૂરી છબીઓ, બેકગ્રાઉન્ડ અને લિન્કનો ઉપયોગ કરો. તમે લખાણને સુધારી પણ શકો. ઉદાહરણ તરીકે, હોમપેજ પર તમે તમારી શાળાનું નામ, શાળાના ટ્રસ્ટનું નામ, શાળાનો સ્થાપના દિન, શાળાના સંપર્ક માટેની માહિતી વગેરે લખી શકો.

સારાંશ

આ પ્રકરણમાં HTML દસ્તાવેજનાં head વિભાગમાં ઉપયોગમાં લેવાતા મેટા-ટેગ (meta-tag) વિશે શીખ્યા. મેટા-ટેગ એમની રીતે કંઈ પણ દર્શાવે નહીં, તેમ છતાં તે લેખક, ચાવીરૂપ શબ્દો, પાના વિશેનું વર્ણન સંબંધી માહિતી પ્રદાન કરે છે. આવી માહિતી સર્ચએન્જિન તથા અન્ય યુટિલિટી પ્રોગ્રામને ખૂબ જ મદદરૂપ નીવડી શકે છે. Head વિભાગની જેમ, HTML દસ્તાવેજના Body વિભાગમાં ઉપયોગમાં લેવાતા ટેગ અને ગુણધર્મોની પણ ચર્ચા કરી HTMLનું body વિભાગ ટેગની ગોઠવણી સહિત ભૌતિક અને તાર્કિક સ્વરૂપાંકન ટેગ અને ખાસ ચિહ્નો દર્શાવવાની સવલત પણ આપે છે. આપણે HTML દસ્તાવેજમાં ટિપ્પણી સર્વસામાન્ય (ડિફોલ્ટ) ફોન્ટ અને રંગ ઉમેરવાની રીત જોઈ. અંતમાં, આપણે એક નમૂનારૂપ શાળાની વિવિધ પ્રવૃત્તિઓ વર્ણવતા ઘણા વેબપેઈજઓ ધરાવતી એક નાની વેબસાઈટ પણ બનાવી.

સ્વાધ્યાય

1. કોઈ HTML દસ્તાવેજ માટે head વિભાગ અને તેમાં ઉપયોગમાં લેવાતા મેટા-ટેગ વિશે ટૂંક નોંધ લખો.
2. HTML ટેગનો ઉપયોગ કરી કેવી રીતે કોઈ છબીને વેબપેઈજ પર બેકગ્રાઉન્ડ તરીકે ઉમેરી શકાય, તે વર્ણવો.
3. ઇલેક્ટ્રોનિક માધ્યમોમાં રંગોની રજૂઆત વર્ણવો. “ક્લર કોડિંગ સ્કીમ” વિશે પણ ટૂંકમાં વર્ણવો.
4. HTMLમાં આડી લીટી માટેના <hr /> એલિમેન્ટની ચર્ચા કરો.
5. HTMLમાં font ટેગ વર્ણવો.
6. HTML દ્વારા ખાસ ચિહ્નો કેવી રીતે રજૂ કરાશે તે વર્ણવો.
7. નીચે આપેલ પ્રશ્નો માટે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ આપો:

(1) નીચેનામાંથી કયા બે, HTML કોડના બે મૂળભૂત વિભાગો બનાવે છે ?

- | | |
|----------------------|--------------------------|
| (a) Head અને Body | (b) Physical અને Logical |
| (c) Code અને Browser | (d) Meta-tags અને Body |

(2) HTML દસ્તાવેજમાં નીચેનામાંથી કયા વિભાગમાં મેટા-ટેગ લખવામાં આવે છે ?

- | | |
|----------|-------------|
| (a) Body | (b) Code |
| (c) Head | (d) Special |

(3) HTML પાનામાં જ્યારે મેટા-ટેગ ઉપયોગમાં લેવાય ત્યારે, વેબપેઈજ વિશે નીચેનામાંથી શું વર્ણવવામાં આવે છે ?

- | | |
|--------------------------------------|------------|
| (a) લેખક, ઉદ્દેશ્ય અને ચાવીરૂપ શબ્દો | (b) માળખું |
| (c) સ્ટાઈલ | (d) કદ |

(4) વેબપેઈજનું શીર્ષક નીચેનામાંથી કયા ટેગમાં આવરીને દર્શાવાય છે ?

- | | |
|--------------------------|------------------------|
| (a) <p> અને </p> | (b) <body> અને </body> |
| (c) <title> અને </title> | (d) <h1> અને </h1> |

(5) HTML દસ્તાવેજમાં નીચેનામાંથી કયા વિભાગમાં આપણે ટિપ્પણી ઉમેરી શકીએ ?

- | | |
|------------------------|--------------------|
| (a) Head | (b) Body |
| (c) Head અને body બંને | (d) Head અથવા body |

(6) HTML કોડમાં રંગ દર્શાવવા માટે નીચેનામાંથી શું ઉપયોગમાં લેવાય છે ?

- | | |
|------------------------------|----------------------------|
| (a) સોળઅંકી પદ્ધતિમાં કલરકોડ | (b) દશાંશ પદ્ધતિમાં કલરકોડ |
| (c) કલર મિક્સિંગ મોડેલ | (d) ટકાવારીમાં પિક્સેલ |

(7) HTML કોડમાં મુલાકાત લેવાઈ ગયેલ (visited) લિન્ક ગોઠવવા નીચેનામાંથી શું ઉપયોગમાં લેવાય છે ?

- | | |
|-----------------|----------------|
| (a) a:link | (b) v:link |
| (c) before link | (d) after link |

(8) HTML કોડમાં એક્ટિવ લિન્ક ગોઠવવા માટે નીચેનામાંથી શું ઉપયોગમાં લેવાય છે ?

- (a) alink (b) vlink
(c) before link (d) after link

(9) HTMLમાં અક્ષરોના સ્વરૂપાંકન માટેના નીચેનામાંથી કયા બે મુખ્ય પ્રકારો છે ?

- (a) ભૌતિક અને તાર્કિક (b) આંતરિક અને બાહ્ય
(c) કામચલાઉ અને કાયમી (d) Head અને Body

પ્રાયોગિક સ્વાધ્યાય

1. પ્રકરણમાં ચર્ચા કરેલ 'My School' વેબસાઇટ બનાવવાનો પ્રોજેક્ટ પૂરો કરો.
2. અન્ય શૈક્ષણિક સંસ્થાઓની વેબસાઇટ જુઓ અને તેની રચનાનો અભ્યાસ કરો. આ રચનાઓ માટે ટિપ્પણીઓ લખો.
3. તમારા વિશેનું વેબપેઈજ તૈયાર કરો. તમારા વિશે, તમારા પરિવાર વિશે, તમારા મિત્રો, તમારા શૈક્ષણિક ધ્યેય અને સંપર્ક માટે સરનામાનું વર્ણન કરો. વેબપેઈજની રચના તમે નીચે મુજબ વિચારી શકો.

I, me and myself...!

[About Me](#)

[My Family](#)

[My Friends](#)

[My Goal](#)

[Contact Me](#)

.....

..... you may write about yourself here....

.....

અહીં બીજી લીટી પર નીચે લીટી દોરીને દર્શાવેલ શબ્દો એ હાઈપર લિન્ક છે, જે તમને જુદા પાન પર સાંકળશે. સક્રિય (એક્ટિવ) લિન્ક અને મુલાકાત લેવાઈ ગયેલ (visited) લિન્ક ગોઠવો. આછા રંગનું રંગીન બેકગ્રાઉન્ડ બનાવો. જે રીતે અગાઉ આપણે 'My School' નામની વેબસાઇટ બનાવી તે પ્રમાણે આ એક સંપૂર્ણ વેબસાઇટ બનવી જોઈએ.

4. <http://www.w3schools.com/> પર જાઓ. મેનુમાંથી HTML વિભાગ શોધી કાઢો અને "try it yourself" ને ચકાસી જુઓ. આ એક એડિટર છે જે તમારા HTML કોડને લઈને તે જાણે કે બ્રાઉઝરમાં નિહાળતા હોય તેવું દૃશ્ય રજૂ કરે છે.



HTMLમાં છબીઓનું વ્યવસ્થાપન

એવું કહેવાય છે કે, હજાર શબ્દો કરતાં એક ચિત્ર વધુ કહી જાય છે. ચિત્ર અથવા છબી મોટાભાગના વ્યક્તિઓને આકર્ષે છે અને પ્રેક્ષકોને અગત્યનો સંદેશ પહોંચાડે છે. જો વેબપેઈજમાં ચિત્ર કે છબી દર્શાવાશે તો ઉપયોગકર્તા તરત જ તેના પર ધ્યાન આપશે. વળી, એવું અવલોકન છે કે સામાન્ય રીતે ચિત્ર કે છબી સમજ અને સ્વીકૃતિની માત્રામાં વધારો કરે છે.

HTMLમાં ટેગનો ઉપયોગ કરીને વેબપેઈજમાં છબીઓ ઉમેરવામાં આવે છે. અગાઉના પ્રકરણમાં આપણે પણ બેકગ્રાઉન્ડ તરીકે એક છબી ઉમેરી છે. વેબપેઈજ ઉપર છબી ઉમેરવાના બીજા પણ ઘણા રસ્તા છે. ઉદાહરણ તરીકે, નીચેની સૂચના વેબપેઈજમાં rainbow.jpg નામની છબી ઉમેરશે.

ઉપરોક્ત ઉદાહરણમાં src ગુણધર્મનો ઉપયોગ કરાયો છે, જેનું નીચે ટેગ સાથે ઉપયોગમાં લઈ શકાતા અન્ય ગુણધર્મોની સાથે વર્ણન કરવામાં આવ્યું છે.

image એલિમેન્ટ(image ટેગ, છબી અને ગુણધર્મો સાથે)માં નવી લીટી (line break) આવતી નથી. માટે તે ઇનલાઈન એલિમેન્ટ (Inline Element) તરીકે ઓળખાય છે.

src ગુણધર્મ

src ગુણધર્મ છબીના સ્થાન અંગેની માહિતી પૂરી પાડે છે. અહીં સ્થાન એટલે છબીનું ઉદ્ગમસ્થાન. તે બ્રાઉઝરને જણાવે છે કે છબીની ફાઈલ ક્યાં પડેલી છે, એટલે કે તેને ક્યાં શોધવી. ઘણીવાર ઇમેજના સ્થાન માટે URLનો ઉપયોગ કરીએ છીએ. છબીનું મૂળ સ્થાન દર્શાવ્યા વગર બ્રાઉઝર માટે છબીને શોધવી અને દર્શાવવી અશક્ય બની જાય છે. "http://pritisajja.info/images/img1.jpg" તરીકે આપેલ URL પરથી છબીને સામેલ કરવા આપણે નીચે મુજબની ટેગ લખી શકીએ.

આ રીતે દર્શાવવામાં આવેલ છબી ફાઈલ યોગ્ય છબી સ્વરૂપ જેમ કે, bmp, gif, tiff અથવા jpg સ્વરૂપની હોવી જોઈએ.

એક એવી પરિસ્થિતિની કલ્પના કરો કે, જેમાં તમે ઇન્ટરનેટમાં માહિતી જોઈ રહ્યા હો અને તમને કોઈ છબી ગમી જાય છે. તો કોઈ HTML કોડમાં તેને સામેલ કરવા તેનું સંપૂર્ણ URL તમે કેવી રીતે મેળવશો? જવાબ એકદમ સરળ છે. પ્રથમ વિકલ્પ છે, તે છબી પર માઉસનું જમણું બટન ક્લિક કરી image linkની કૉપી કરવી અને બીજો વિકલ્પ છે જમણું બટન ક્લિક કરી છબીને તમારા કમ્પ્યુટરની હાર્ડડિસ્કમાં સાચવી લેવી. બીજા વિકલ્પ મુજબ અમલ કરતાં પહેલાં છબીના કૉપીરાઈટ બાબતે સચેત રહેવું જરૂરી છે.

જ્યારે લખાણ સાથે <p> અથવા <h1> જેવા ટેગનો ઉપયોગ કરાય ત્યારે આ ટેગનું કામ માત્ર લખાણને ચોક્કસ ઢબે દર્શાવવાનું હોય છે, જ્યારે image ટેગનો ઉપયોગ કરવામાં આવે ત્યારે, આપણે કોઈ લખાણ આપવાનું હોતું નથી, માત્ર છબીનું મૂળ સ્થાન અને સાથે કેટલાક જરૂરી ગુણધર્મો જ દર્શાવવાના હોય છે. એટલા માટે જ image ટેગને ખાલી ટેગ કહે છે. જો કે, જ્યારે બ્રાઉઝરમાં છબી ધરાવતું કોઈ HTML દસ્તાવેજ દર્શાવાય ત્યારે બ્રાઉઝરે તે છબીને પ્રાપ્ત કરવી જરૂરી છે. છબી તમારા સ્થાનિક કમ્પ્યુટર અથવા સર્વરમાં ઉપલબ્ધ હોવી જરૂરી છે. જો તમે URLનો ઉપયોગ કર્યો હોય તો તમારું ઇન્ટરનેટ જોડાણ ચાલુ છે કે કેમ તે ચકાસો.

છબીઓ માટે એક અલાયદું ફોલ્ડર કે ડિરેક્ટરી બનાવવી એ એક સારી વ્યવસ્થા ગણાય. આ અભિગમ બહેતર છે, ખાસ કરીને જ્યારે વેબસાઈટ ઘણી મોટી હોય અને અનેક છબીઓનો ઉપયોગ કરતી હોય. HTMLમાં એક પછી એક ત્રણ છબીને ઉમેરવા માટેનો કોડ કોડ-લિસ્ટિંગ 3.1માં દર્શાવેલ છે.


```

<html>
<head>
    <title>My favourite food ....!</title>
</head>

<!-- ----- -->
<body>
    <h1> My favourite food ....!</h1>

    <!-- ----- -->
    <h1> <u> Chocolates </u></h1>

    <p>
    
    <h1>
    Chocolates are good for health, better for hunger and best for mood !
    Do not forget to clean your teeth until it is too late !
    </h1>
    </p>

    <!-- ----- -->
    <h1><u> Fruits and Dry Fruits </u></h1>
    <p>
    
    <h1>
        Do not think dry fruits are dry, though they are dried fruits !
        They are really really very interesting !
    </h1>
    </p>

    <!-- ----- -->
    <h1> <u> Ice creams </u> </h1>

    <p>
    
    <h1>
        Ice creams are really cool ! These become coolest when you taste them !
    </h1>
    </p>

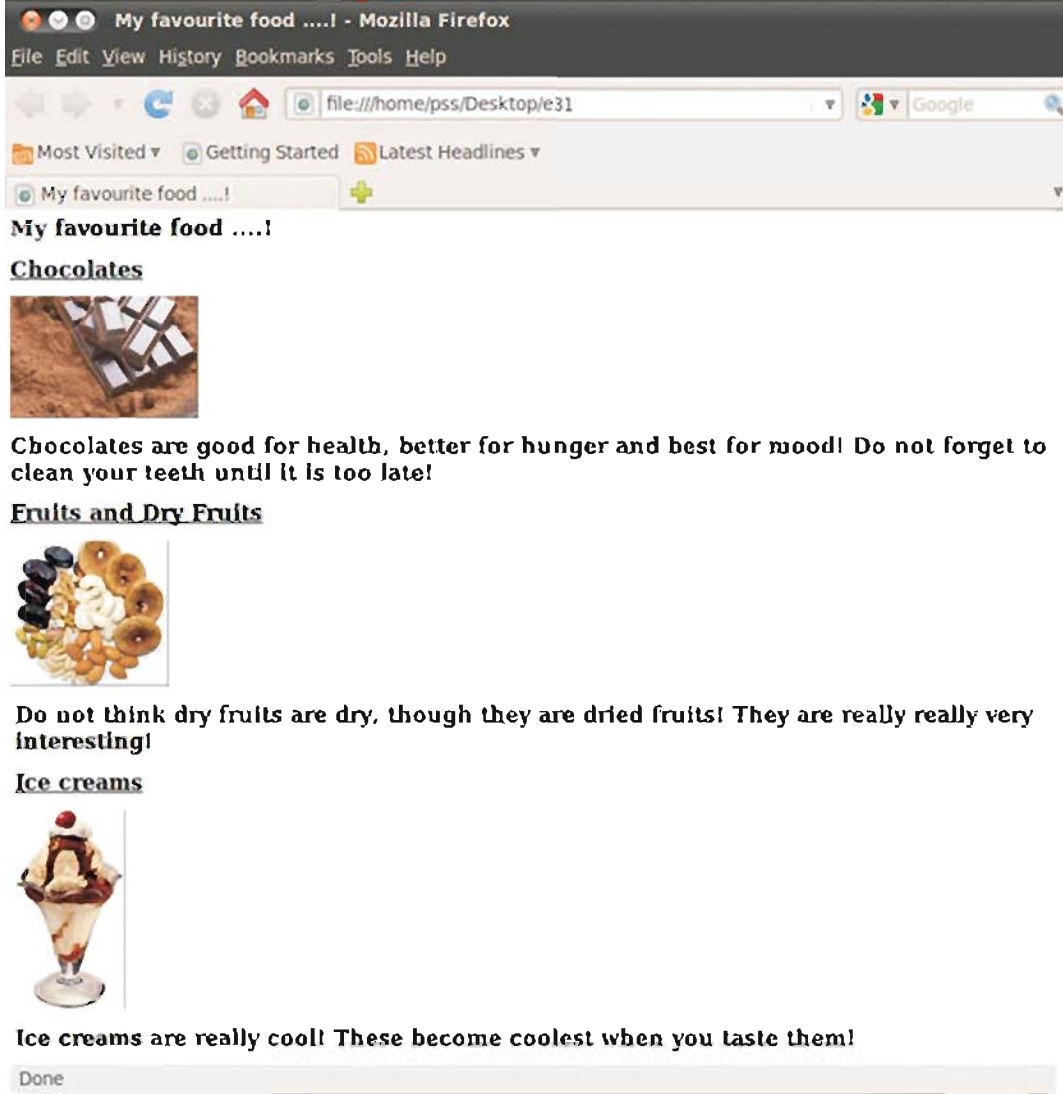
    <!-- ----- -->
</body>
</html>

```

કોડ-વિસ્ટિંગ 3.1 : વેબપેઈજ પર અનેક છબીઓ ઉમેરવી

કોડ-લિસ્ટિંગ 3.1 કેટલીક ટપકાંવાળી લીટીઓ ધરાવે છે. આ ટપકાંવાળી લીટીઓ વેબપેઈજમાં દર્શાવાશે નહીં, કારણ કે, તેને કોમેન્ટ ટેગ (<!-- અને -->) માં આવરીને દર્શાવાઈ છે. આવી લીટીઓ HTML કોડને વાંચવામાં સરળતા બક્ષવા અને વધુ સારી રીતે સમજવા સ્વતંત્ર વિભાગોમાં વિભાજિત કરે છે. બ્રાઉઝર માત્ર માન્ય HTML વિધાનો જ સ્વીકારતું હોવાને કારણે આપણે ટપકાંવાળી લીટીને સીધેસીધી ટાઈપ કરી શકતાં નથી. આપણે આવી ટિપ્પણી જેવી લીટીઓને કોમેન્ટ ટેગમાં આવરીને દર્શાવવી જોઈએ.

કોડ-લિસ્ટિંગ 3.1માં આપેલ HTML કોડનું પરિણામ આકૃતિ 3.1માં દર્શાવાયું છે.



આકૃતિ 3.1 : વેબપેઈજમાં અનેક છબીઓ

બ્રાઉઝર આપેલા સ્થાન (source) પરથી એક પછી એક છબીઓ મેળવશે અને HTML કોડમાં જે ક્રમ દર્શાવ્યો હશે તે પ્રમાણે તેને ક્રમમાં દર્શાવશે. ઉપયોગકર્તા સામાન્ય રીતે તમામ છબીઓ સહિત પાનાને નિહાળે છે.

alt ગુણધર્મ

image ટેગ સાથે alt ગુણધર્મ મૂળ છબીને વર્ણવે છે. આ ગુણધર્મને મૂકવો એ હંમેશાં સલાહભર્યું છે, કારણ કે તે છબીને વર્ણવે છે. alt ગુણધર્મનું ઉદાહરણ નીચે મુજબ છે:

આ ગુણધર્મ અત્યંત ઉપયોગી છે. તેનું પહેલું કારણ એ છે કે, તે છબીને વર્ણવે છે, જે ઘણી બધી વખત જરૂરી હોય છે. અન્ય કારણ એ છે કે, ધારો કે, કોઈ પણ કારણોસર બ્રાઉઝર છબીને ન દર્શાવે તો પણ વાચક એટલું તો જાણતો જ હશે કે કયું ચિત્ર આવશે. ત્રીજું કારણ એ છે કે, જો વેબપેઈજ 'સ્ક્રીન રીડર' (Screen Reader)

જેવા સોફ્ટવેર દ્વારા વાંચવામાં આવતું હશે તો છબીનું અર્થઘટન સરળ બની જશે. (સ્ક્રીન રીડર એ દષ્ટિની ખામી ધરાવતા લોકો માટે તૈયાર કરાયેલું એક વિશિષ્ટ સોફ્ટવેર છે)

height અને width ગુણધર્મ

height અને width ગુણધર્મ નીચે દર્શાવ્યા પ્રમાણે છબીની ઊંચાઈ અને પહોળાઈ કેટલા પિક્સેલ છે તે દર્શાવે છે.

અહીં, width ગુણધર્મ બ્રાઉઝરને જણાવે છે કે, વેબપેઈજ પર છબી કેટલી પહોળી દેખાવી જોઈએ. height ગુણધર્મ એ દર્શાવે છે કે છબી કેટલી ઊંચી હોવી જોઈએ. height અને width ગુણધર્મની કિંમત પિક્સેલની સંખ્યાને આધારે દર્શાવવામાં આવે છે. height અને width બન્ને એકસાથે ઉપયોગમાં લેવા જરૂરી નથી, તમે બેમાંથી કોઈ પણ એકનો ઉપયોગ કરી શકો. જો કે, કોઈ એક જ ગુણધર્મનો ઉપયોગ કરવાનું કંઈ ખાસ ઉપયોગી પણ નથી.

બ્રાઉઝરને છબીની લંબાઈ અને પહોળાઈ વિશે વધારાની માહિતી પૂરી પાડવાથી બ્રાઉઝરમાં છબીને લઈ આવવાનું (Load કરવાનું) અને રજૂઆત કરવાનું કાર્ય સરળ અને કાર્યક્ષમ બનશે. અહીં, બ્રાઉઝર જાણે છે કે, છબી માટે કેટલી જગ્યા ફાળવવી પડશે. અત્રે એ નોંધવું જોઈએ, કે તમારી છબી તમારા સ્ક્રીન કરતાં મોટી ન હોવી જોઈએ. જો કે, HTML કોડમાં મોટી છબીને અનુકૂળ રીતે સમાવવાનું શક્ય છે. વધુ ઊંચી કે નીચી ઊંચાઈ કે પહોળાઈની કિંમત આપીને તમારી છબીને વિસ્તૃત કરવી કે સંકોચવી એ પણ સલાહભર્યું નથી. આમ કરવાથી છબીની ગુણવત્તા અને દૃશ્યમાનતા નાશ પામશે.

align ગુણધર્મ

align ગુણધર્મ આપેલી છબીને પાનાની અંદર જ અથવા છબી ધરાવતા કોઈ પણ એલિમેન્ટમાં ગોઠવે છે. આ માટેનું ઉદાહરણ નીચે મુજબ છે :

align ગુણધર્મની સાથે કોષ્ટક 3.1માં દર્શાવ્યા મુજબની કોઈ એક કિંમત આપી શકાય છે.

કિંમત	ઉદ્દેશ્ય
Top	છબી લખાણની હાલની લીટીના ઉપરના ભાગે ગોઠવાય છે.
Middle	છબી એ રીતે ગોઠવાય છે કે, છબીનો મધ્યભાગ હાલના લખાણ પાસે દેખાય છે.
Bottom	છબીનો છેક નીચેનો ભાગ લખાણની હાલની લીટી સાથે ગોઠવાય છે.
Left	છબી વિન્ડોની ડાબી બાજુ ગોઠવાય છે.
Right	છબી વિન્ડોની જમણી બાજુ ગોઠવાય છે.

કોષ્ટક 3.1 : align ગુણધર્મની શક્ય કિંમતો

border ગુણધર્મ

વેબપેઈજમાં છબીની ફરતે કિનારી દોરીને છબીને વધુ આકર્ષક બનાવી શકાય છે. સામાન્ય રીતે છબીઓને ચારબાજુ ફરતે કિનારી હોતી નથી.

છબીની ફરતે કિનારી બનાવવા માટે નીચે મુજબનો border ગુણધર્મ ઉપયોગમાં લઈ શકાય. border ગુણધર્મમાં borderની પહોળાઈ પિક્સેલમાં દર્શાવાય છે.

જ્યારે છબીને જોડાણ અપાયું હોય ત્યારે કેટલાંક બ્રાઉઝર (જેવાં કે, Internet Explorer) છબીને વધુ પ્રકાશિત (highlight) કરે છે. એટલે કે, જ્યારે છબી જોડાણ તરીકે ઉપયોગમાં લેવાઈ હશે ત્યારે તે કિનારી દ્વારા વધુ પ્રકાશિત કરાશે.

id ગુણધર્મ

id ગુણધર્મની મદદથી, તમે છબી માટે કોઈ ઓળખ (identifier) એટલે કે નામ આપી શકો. એ પછી જાવા (java) જેવી પ્રોગ્રામિંગ ભાષામાં લખાતી સ્ક્રિપ્ટમાં તે છબીને તે નામથી સંબોધી શકાય. id ગુણધર્મનો ઉપયોગ કેવી રીતે કરાય તે નીચેની ટેબલ દ્વારા દર્શાવાયું છે :

ઈમેજની ચારેબાજુ ખાલી જગ્યા ઉમેરવી (Adding space around image)

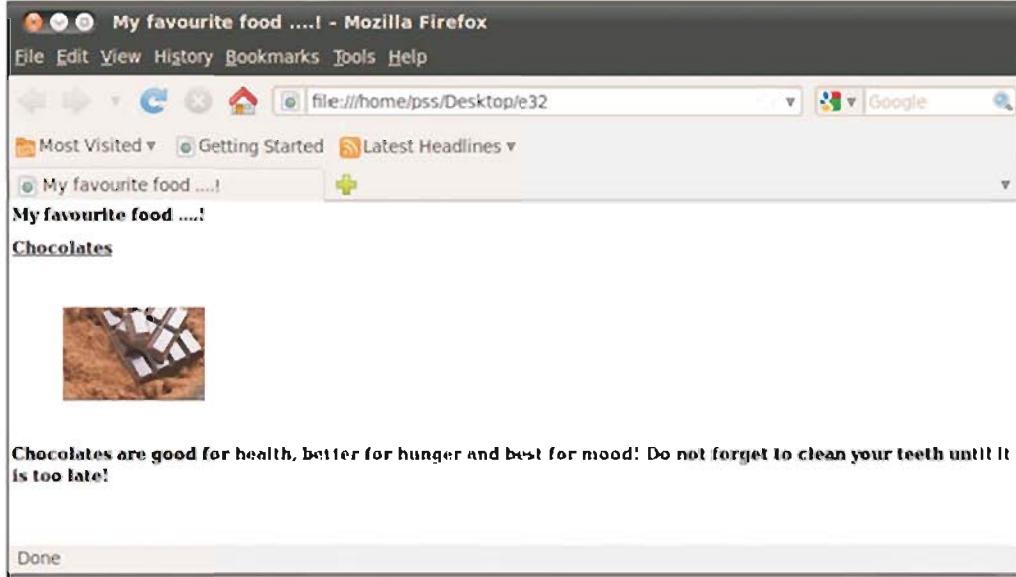
ઈમેજની ચારેબાજુ ખાલી જગ્યા ઉમેરવા માટે `vspace` અને `hspace` ગુણધર્મનો ઉપયોગ કરવામાં આવે છે. છબીની ઉપર અને નીચેના ભાગે ખાલી જગ્યા ઉમેરવા `vspace` ગુણધર્મનો ઉપયોગ કરવામાં આવે છે એ જ રીતે, છબીની ડાબી અને જમણી બાજુએ ખાલી જગ્યા દર્શાવવા આ બન્ને ગુણધર્મોનો ઉપયોગ નીચેના ઉદાહરણમાં દર્શાવાયો છે:

જ્યારે કોઈ છબી લખાણની બિલકુલ અડોઅડ હોય અને જરાય જગ્યા ન દેખાતી હોય ત્યારે આ ગુણધર્મ ખૂબ જ ઉપયોગી નીવડે છે. છબીની ચારેય બાજુ જગ્યા છોડવાથી છબી જોવામાં વધુ સ્પષ્ટ બને છે અને વેબપેઈજ વધુ વાંચવા યોગ્ય બને છે. કોડ-લિસ્ટિંગ 3.2 એક એવું ઉદાહરણ દર્શાવે છે કે જેમાં ચોકલેટની છબીની ફરતે થોડી જગ્યા છોડવામાં આવી છે.

```
<html>
<head>
  <title>My favourite food ....!</title>
</head>
<!-- ----- -->
<body>
  <h1> My favourite food ....!</h1>
  <!-- ----- -->
  <h1> <u> Chocolates </u></h1>
  <p>
    
  </p>
  <h1>
    Chocolates are good for health, better for hunger and best for mood !
    Do not forget to clean your teeth until it is too late !
  </h1>
</p>
<!-- ----- -->
</body>
</html>
```

કોડ-લિસ્ટિંગ 3.2 : hspace અને vspace વર્ણવતો HTML કોડ

જ્યારે તમે કોડ-લિસ્ટિંગ 3.2માં દર્શાવેલ HTML કોડને બ્રાઉઝરમાં નિહાળશો તો તે આકૃતિ 3.2 જેવો દેખાશે. hspace અને vspace ગુણધર્મ દ્વારા દર્શાવવામાં આવતી આડી અને ઊભી જગ્યા જોઈ શકાય છે. HTML કોડમાં થોડી વધુ છબીઓ ઉમેરો અને પછી જુઓ કે તમારું બ્રાઉઝર તે છબીઓને કેવી રીતે રજૂ કરે છે.



આકૃતિ 3.2 : hspace અને vspaceનું નિદર્શન

હવે તમે એ જોધું કે, hspace અને vspace ગુણધર્મ છબીની બન્ને બાજુ જગ્યા ઉમેરે છે. એટલે કે, જ્યારે hspaceનો ઉપયોગ કરવામાં આવે ત્યારે તે છબીની ડાબી અને જમણી એમ બન્ને બાજુ જગ્યા છોડે છે. એ જ રીતે, જ્યારે vspace ગુણધર્મનો ઉપયોગ કરવામાં આવે છે ત્યારે તે છબીની ઉપરની અને નીચેની બાજુએ જગ્યા છોડે છે. માત્ર કોઈ એક તરફ જ છબી ઉમેરાય તેવું કરવા માટે આપણે છબીને યોગ્ય રીતે સુધારવી પડે. એના વિકલ્પરૂપે મૂળ છબીની બાજુમાં એક ખાલી છબી છાપવાનું પણ વિચારી શકાય. જો કે, આ બીજા વિકલ્પમાં જગ્યા અને ગોઠવણી કરવાનું થોડું અઘરું બની જાય.

તમારા કમ્પ્યુટરની સ્મૃતિમાંથી "chocolate.png" ફાઈલને દૂર કરવાનો પ્રયાસ કરો અને જુઓ કે શું થાય છે ? શું તે છબી માટે વૈકલ્પિક વર્ણન જેમ કે, "Here comes the sweet image" એવું દર્શાવશે ? તમને કદાચ આકૃતિ 3.3માં દર્શાવ્યા મુજબનું પરિણામ જોવા મળે.



આકૃતિ 3.3 : છબીની અનુપસ્થિતિમાં વૈકલ્પિક વર્ણન

ઓછી રિઝોલ્યુશનવાળી છબી

વધુ રિઝોલ્યુશનવાળી છબી ઘણી વધુ જગ્યા રોકે છે અને વેબપેઈજમાં રજૂ થવામાં ઘણી વાર લગાડે છે. વધુ રિઝોલ્યુશનવાળી છબી વેબપેઈજમાં ધીમે ધીમે રજૂ થાય ત્યાં સુધી આપણે કામચલાઉ ઓછા રિઝોલ્યુશનવાળી છબીને નીચે દર્શાવ્યા મુજબ વેબપેઈજમાં રજૂ કરી શકીએ :

કોઈ પણ છબીને ઓછા રિઝોલ્યુશનવાળી છબી બનાવવા માટે તમે છબીમાં સુધારા-વધારા કરવાની સવલત આપતા 'ઇમેજ એડિટિંગ ટૂલ'ની resize, crop વગેરે ટેકનિકનો ઉપયોગ કરી શકો.

ઓછા રિઝોલ્યુશનવાળી છબી મૂકવાના વિકલ્પના ગુણધર્મના ઉપયોગ દ્વારા આપણે બ્રાઉઝરની ગતિની સમસ્યા ઉકેલી શકીએ તેમ છતાં, આપણે વૈકલ્પિક છબી બનાવવી પડે અને તેને સાચવવી પડે. વળી, ઓછા રિઝોલ્યુશનવાળી છબી ધૂંધળી અને અસ્પષ્ટ દેખાશે.

વિવિધ સ્વરૂપોની છબીઓને સમાવવી

અત્યાર સુધી આપણે Image ટેગમાં bmp (Bit Mapped Picture) ફાઈલનો જ સમાવેશ કર્યો. પરંતુ, છબીના થોડાં અન્ય સ્વરૂપો પણ ઉપલબ્ધ છે, જેનો Image ટેગમાં સમાવેશ કરી શકાય. છબી માટેનાં વિવિધ સ્વરૂપો પૈકી bmp, jpeg, png, tiff અને gif એ વધુ પ્રચલિત છબી સ્વરૂપો છે. આવાં છબીસ્વરૂપોનું વર્ણન કોષ્ટક 3.2માં દર્શાવેલ છે.

છબી ફાઈલ સ્વરૂપ	વર્ણન
BMP	Windows અને OS/2 માટેની Bitmap Graphic file
GIF	Graphic Interchange Format file
JPEG	Joint Photographic Expert Group file
JPG	JPEG/JIFF Image file
PNG	Portable Network Graphics file
TIFF	Tag Image File Format

કોષ્ટક 3.2 : છબી સ્વરૂપો

ડિજિટલ કેમેરા અને વેબપેઈજ સામાન્ય રીતે jpeg/jpeg ફાઈલોનો ઉપયોગ કરે છે. કારણ કે, આ સ્વરૂપો ફાઈલમાં ડેટાનું નોંધપાત્ર રીતે સંકુચન કરે છે. ફોટોગ્રાફ જેવી (continuously toned) છબી માટે jpeg અને jpeg છબી સ્વરૂપ આદર્શ છે. વ્યાપારી ધોરણે, tiff છબીસ્વરૂપ વધુ ઉપયોગમાં લેવાય છે, કારણ કે તે ઉચ્ચ ગુણવત્તા અને લઘુત્તમ નુકસાન સાથે છબીનું સારા એવા પ્રમાણમાં સંકુચન કરી આપે છે. Tiff એ ખરેખર નુકસાનરહિત સારા છબી સ્વરૂપોમાં એક છબીસ્વરૂપ છે. gif અને png જેવા અન્ય ફાઈલ સ્વરૂપો પણ નુકસાનરહિત સંકુચન ટેકનોલોજીનો ઉપયોગ કરે છે. જો કે, png સ્વરૂપ લખવા અને વાંચવા માટે પ્રમાણમાં ધીમું છે.

આકર્ષણ કેન્દ્ર હોટ સ્પોટ તરીકે છબી

ઘણી વખત પ્રમાણમાં મોટી છબીને દર્શાવવામાં બ્રાઉઝર થોડો સમય લગાડે છે. આ ઉપરાંત જ્યારે આપણી પાસે એક કે વધુ પાનાં પર અનેક છબીઓ હોય ત્યારે ઉપયોગકર્તા માટે મૂંઝવણરૂપ બને છે કે કઈ છબીને પ્રથમ ગણવી. શું મોબાઈલ ફોન, ઘરેણાં, પુસ્તકો અને એના જેવી ચીજવસ્તુઓ માટેની ઓનલાઈન ખરીદી માટેની વેબસાઈટ પર તમારું ધ્યાન પકડ્યું છે ? એક જ વેબપેઈજ પર તમને કદાચ ઘણી ચીજવસ્તુઓની ટૂંકા વર્ણન સાથેની અનેક નાની છબીઓ જોવા મળશે. જ્યારે તમને એમાંથી કોઈ વસ્તુમાં રસ પડે તો તમે તે નાની છબી પર ક્લિક કરી શકો, જેથી તમને પસંદ પડેલી ચીજવસ્તુ માટે સારી ગુણવત્તાવાળી પ્રમાણમાં મોટી છબીની સાથે સાથે વસ્તુ વિશેની વિસ્તૃત માહિતી ધરાવતું વેબપેઈજ દર્શાવાય. આવું કરવા માટે વેબપેઈજ પરની દરેક નાની છબીને જોડાણ (લિન્ક) આપવી પડે. તમે એવું કહી શકો કે, અહીં છબીએ હોટ ટેક્સ્ટ (hot text) જેવું કાર્ય કરે છે. જોકે, અહીં કોઈ ટેક્સ્ટ તો છે નહીં ! માટે ખરેખર આને હોટ ટેક્સ્ટ તો કહી ન શકાય ! ખરુંને ! તેથી હોટ ટેક્સ્ટ જેવું કામ કરતી આ છબીવાળી જગ્યા(spot)ને હોટ સ્પોટ (hot spot) કહે છે. આવી બે નાની છબી ધરાવતી અને બે જુદા જુદા વેબપેઈજ તરફ દોરી જતા HTML કોડ, કોડ-લિસ્ટિંગ 3.3માં રજૂ કરવામાં આવેલ છે.

```

<html>
  <head>
    <title> Image Hot Spot </title>
  </head>
  <!-- ----->
  <body>
    <h1> Either you can go to office or go to temple!</h1>

    <p> <h2>Click on any image below to see its larger version </h2> </p>

    <br>
  <!-- ----- adding links to the images ----- -->

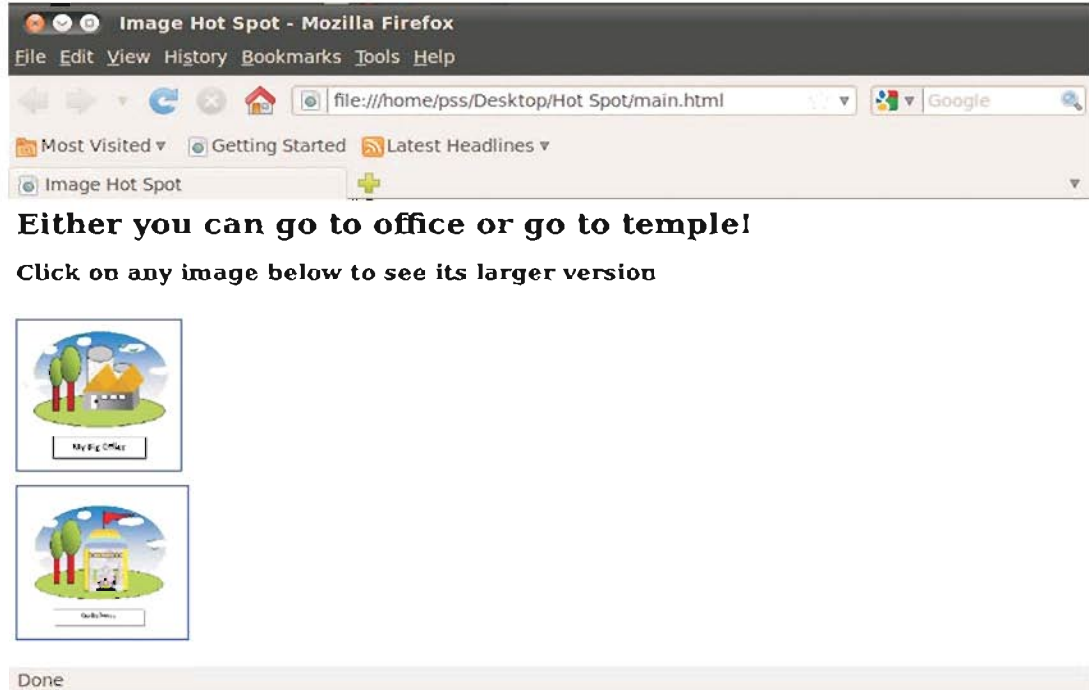
    <a href="big_office.html"></img></a></p>

    <a href="big_temple.html"></img></a></p>
  <!-- ----->
  </body>
</html>

```

કોડ-લિસ્ટિંગ 3.3 : હોટસ્પોટ તરીકે છબી માટેના HTML કોડ

main.html ફાઈલમાં ઉપરોક્ત કોડ ટાઈપ કરી સંગ્રહ કરો. જો તમે આ કોડને બ્રાઉઝર દ્વારા નિહાળશો તો તે આબેહૂબ આકૃતિ 3.4માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 3.4 : હોટ સ્પોટ તરીકે છબીનો ઉપયોગ

કોડ-લિસ્ટિંગ 3.3માં દર્શાવેલ HTML કોડ ઉપરાંત પણ આપણે ચાર છબીની જરૂર પડશે. આપણને small_office.bmp નામની ઓફિસની એક નાની છબીની જરૂર પડશે અને big_office.bmp નામની ઓફિસની એક મોટી છબી ફાઈલની પણ જરૂર પડશે. એ જ રીતે, આપણને small_temple.bmp નામની મંદિરની એક નાની છબી અને big_temple.bmp નામની મંદિરની એક મોટી છબીની પણ જરૂર પડશે. આ માટે તમારે નવી છબીઓ તૈયાર કરવી પડે અથવા હયાત છબીઓનાં નામ બદલીને પણ ઉપયોગમાં લઈ શકો.

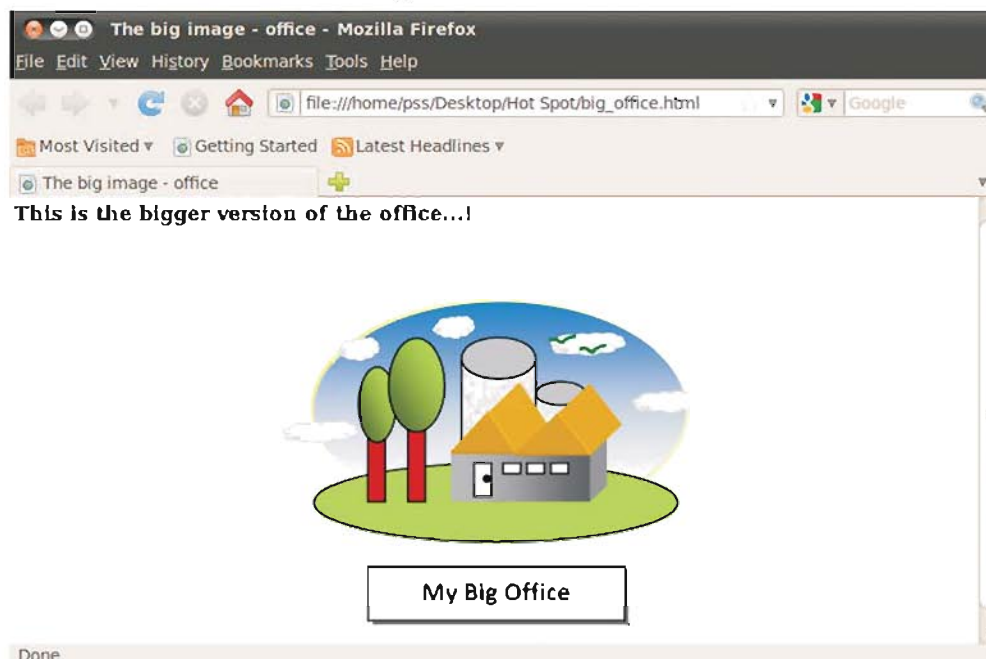
જો ઓફિસ માટે (small_office.bmp) અને મંદિર માટે (small_temple.bmp) ફાઇલ ઉપલબ્ધ હશે અને જો તમે main.htmlને બ્રાઉઝર દ્વારા નિહાળશો, તો આકૃતિ 3.4માં દર્શાવ્યા મુજબનું પરિણામ તમે નિહાળી શકશો.

એકવાર બ્રાઉઝરમાં તમે main.html પાનામાં બંને છબીઓને યોગ્ય રીતે નિહાળી લો, પછી છબીઓ પર ક્લિક કરવાનો પ્રયત્ન કરી શકો. જ્યારે તમે પ્રથમ છબી પર ક્લિક કરશો ત્યારે નાની છબીને હોટ સ્પોટ ગણીને તેની નીચે આપેલ href ટેગ તમને બીજા પાના (big_office.html) તરફ લઈ જશે.

</p>

અને એ નોંધ લો કે, ઓફિસની મોટી છબી સાથે તમારી big_office.html તૈયાર હોવી જ જોઈએ.

પ્રથમ છબી પર ક્લિક કરશો ત્યારે તમને આકૃતિ 3.5માં દેખાય છે તેવી સ્ક્રીન જોવા મળશે.



આકૃતિ 3.5 : જ્યારે તમે પહેલી છબી પર ક્લિક કરો ત્યારે

big_office.html તૈયાર કરવા માટેના HTML કોડ કોડ-લિસ્ટિંગ 3.4માં આપેલ છે.

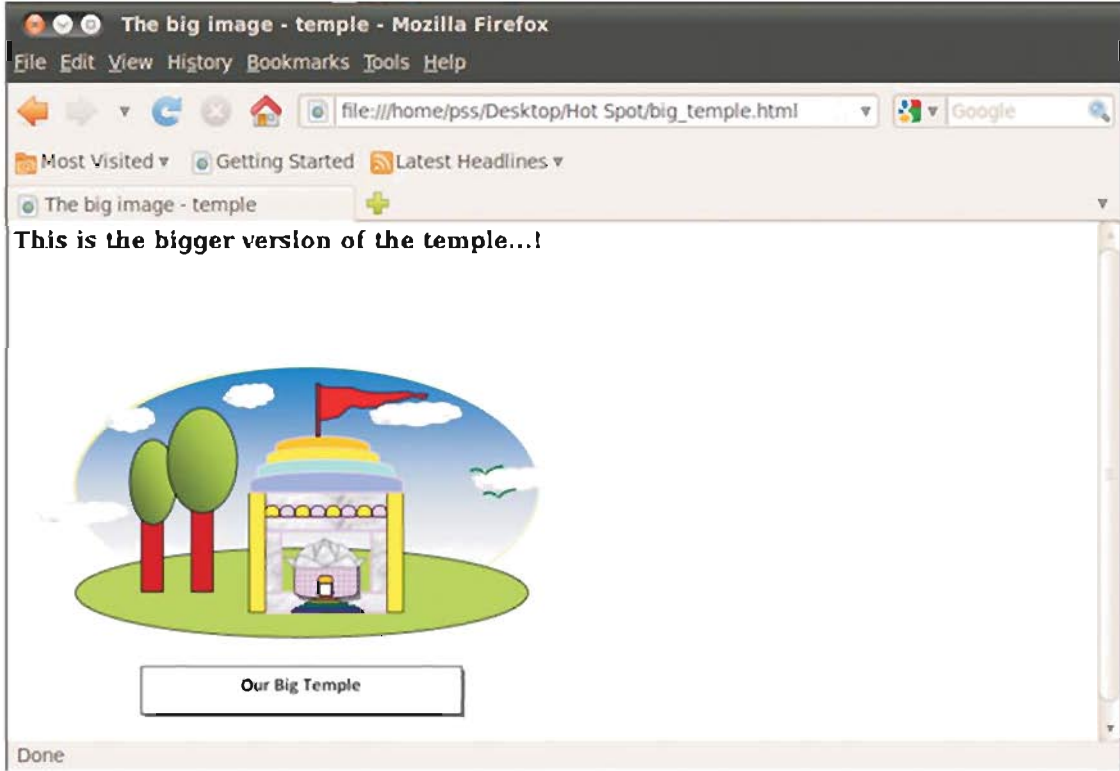
```
<html>
  <head>
    <title> The big image - office </title>
  </head>
  <!-- ----- -->
  <body>
    <h1> This is the bigger version of the office...! </h1>
    <br>
    <br>
    <br>

    </img>

  <!-- ----- -->
</body>
</html>
```

કોડ-લિસ્ટિંગ 3.4 : પ્રથમ છબી માટેના કોડ

એ જ રીતે, જ્યારે આકૃતિ 3.4માં દર્શાવેલી બીજી છબી પર તમે ક્લિક કરશો ત્યારે તમે આકૃતિ 3.6માં દર્શાવ્યા મુજબનું પરિણામ નિહાળી શકશો :



આકૃતિ 3.6 : જ્યારે તમે બીજી છબી પર ક્લિક કરો ત્યારે

આકૃતિ 3.6 માટેનો HTML કોડ (big_temple.html) કોડ-ચિસ્ટિંગ 3.5માં વર્ણવેલ છે.

```
<html>
  <head>
    <title> The big image - temple </title>

  </head>
<!-- _____ -->
<body>
  <h1> This is the bigger version of the temple...! </h1>
  <br>
  <br>
  <br>

  </img>

<!-- _____ -->

  </body>

</html>
```

કોડ-ચિસ્ટિંગ 3.5 : બીજી છબી માટેના કોડ

ઉપરોક્ત ઉદાહરણને પૂર્ણ કરવા તમારે નીચે મુજબ ચાર છબીઓ અને ત્રણ HTML ફાઈલની જરૂર પડી :

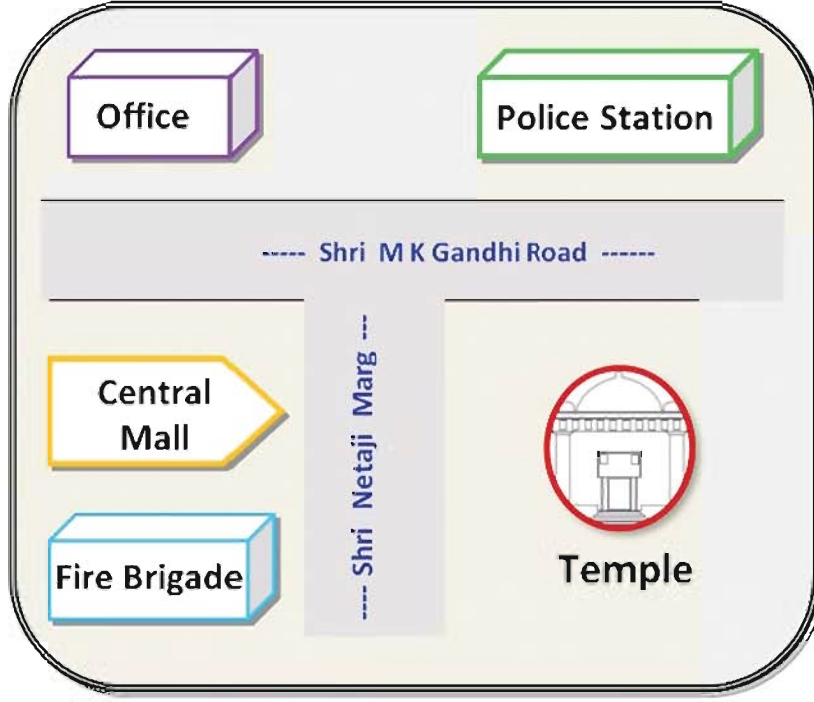
1.	Small_office.bmp	→	bmp સ્વરૂપમાં ઓફિસની એક નાની છબી (અથવા અન્ય કોઈ પણ માન્ય છબી સ્વરૂપ)
2.	Small_temple.bmp	→	bmp સ્વરૂપમાં મંદિરની એક નાની છબી
3.	Big_office.bmp	→	bmp સ્વરૂપમાં ઓફિસની એક મોટી છબી
4.	Big_temple.bmp	→	bmp સ્વરૂપમાં મંદિરની એક મોટી છબી
5.	Main.html	→	બે નાની છબી સાથેની HTML કોડની ફાઈલ, અન્ય બે HTML ફાઈલ સાથે અનુસંધાન અને થોડું લખાણ
6.	Big_office.html	→	ઓફિસની એક મોટી છબી સાથેની HTML કોડની ફાઈલ
7.	Big_temple.html	→	મંદિરની એક મોટી છબી સાથેની HTML કોડની ફાઈલ

મુખ્ય પાનામાં તમે કેટલીક વધુ નાની છબીઓ ઉમેરવાનો પ્રયત્ન કરી શકો. નાની છબીઓને થમ્બનેઇલ (thumbnail) તરીકે પણ ઓળખવામાં આવે છે. જ્યારે તમે વેબપેઈજમાં પદ્ધતિસર ગોઠવાયેલી આવી નાની છબીઓ પર ક્લિક કરશો ત્યારે દરેક છબી તેની વિસ્તૃત (અથવા પ્રમાણમાં મોટી) આવૃત્તિ તરફ દોરી જશે.

Image Map

આગળના વિભાગમાં, આપણે છબી સાથે એક જોડાણ (લિન્ક) કર્યું. એક જ છબી સાથે અનેક જોડાણ કરવાનું પણ શક્ય છે. અહીં દરેક જોડાણ જુદા જુદા સંબંધ તરફ નિર્દેશ કરે છે. એક જ છબીમાં એકથી વધુ જોડાણ બનાવવા આપણે છબીની અંદર એક કરતાં વધુ ક્લિક કરી શકાય તેવાં સ્થાન તૈયાર કરવાં પડશે. આવા દરેક સ્થાનને ‘હોટ સ્પોટ’ (Hot spot) કહેવાય. એવું ધારો કે, તમારી પાસે ધંધાકીય કચેરી (કોર્પોરેટ ઓફિસ), ખરીદ કેન્દ્ર (શોપિંગ મોલ), મંદિર, પોલિસમથક અને અગ્નિશામક દળમથક જેવાં વિવિધ સ્થાનો દર્શાવતો કોઈ વિસ્તારનો નકશો છે. આ દરેક સ્થાન ઉપર આપણે જોડાણ (લિન્ક) બનાવી શકીએ. ઉદાહરણ તરીકે, મંદિરના સ્થાન ઉપર આપણે એવું જોડાણ (લિન્ક) ગોઠવી શકીએ જે આપણને મંદિર વિશેની માહિતી ધરાવતા પાના તરફ લઈ જાય. આ રીતે જોડાણ કરાયેલ વેબપેઈજ કેટલાક ફોટા, સમાચારો, મંદિર વિશેનો ટૂંકો ઇતિહાસ અને મંદિરે કેવી રીતે પહોંચી શકાય તે માટેનું દિશાદર્શન પણ દર્શાવી શકે. જો તમારી પાસે કદાચ કોઈ નકશો (ધારો કે, ભારતનો નકશો) હોય તો નકશાના કેટલાક પ્રદેશો(ગુજરાત, મહારાષ્ટ્ર વગેરે જેવા રાજ્યો)ને ‘હોટ સ્પોટ’ બનાવી શકાય. જ્યારે કોઈ વ્યક્તિ આવા કોઈ હોટ સ્પોટ પર ક્લિક કરે ત્યારે નવું વેબપેઈજ ખોલી શકાય.

આકૃતિ 3.7માં દર્શાવેલી છબીને ધ્યાને લો. છબીમાં એક મંદિર, એક ધંધાકીય કચેરી, એક ખરીદ કેન્દ્ર, એક અગ્નિશામકદળ મથક અને એક પોલિસમથક છે. આ ઉપયોગી કેન્દ્રો “શ્રી એમ. કે. ગાંધી માર્ગ” અને “શ્રી નેતાજી માર્ગ” નામના બે મુખ્ય રસ્તા પર આવેલ છે. આ પાંચેય મકાનો અને બન્ને રસ્તાઓ પર જોડાણ ગોઠવેલ છે. એટલે કે, સાત જુદા જુદા વિસ્તાર પર સાત જોડાણ ગોઠવ્યાં છે. અગાઉ જણાવ્યા અનુસાર જે જગ્યા પર જોડાણ ગોઠવવામાં આવે તે જગ્યાને ‘હોટ સ્પોટ’ કહે છે. હોટ સ્પોટ મોટા અને સ્પષ્ટ દેખાય તેવા હોવા જોઈએ, જેથી ઉપયોગકર્તા આસાનીથી તેમને ઓળખી શકે અને તેના પર ક્લિક કરી શકે. અન્યથા હોટ સ્પોટ પસંદ કરવામાં અને જોડાણને અનુસરવામાં ઉપયોગકર્તાને તકલીફ પડશે. વધુમાં, છબી એવી માહિતી રજૂ કરતી હોવી જોઈએ કે, આવી જગ્યા / હોટ સ્પોટ ઉપર ક્લિક કરવાથી ઉપયોગકર્તાને પસંદ કરેલ જગ્યા વિશે વિસ્તૃત માહિતી દર્શાવતા નવા વેબપેઈજ તરફ લઈ જવામાં આવશે. ઉદાહરણ તરીકે જ્યારે આકૃતિ 3.7માં દર્શાવેલી છબી બ્રાઉઝરમાં દર્શાવાય ત્યારે ઉપયોગકર્તાને “click on temple to know more...” એવો સંદેશ બતાવી શકાય. છબી બનાવી શકતા યોગ્ય “ઈમેજ ક્રિએશન ટૂલ”નો ઉપયોગ કરી આકૃતિ 3.7માં દર્શાવ્યા મુજબની છબી તૈયાર કરો અને તેને city.bmp એવું નામ આપો.



આકૃતિ 3.7 : શહેરના બજારનું દૃશ્ય

હોટ સ્પોટના સ્થાન છબીના ઉપરની બાજુના ડાબા ખૂણેથી x અને y પ્રતિષ્ઠાપકો (co-ordinates) માપીને દર્શાવવામાં આવે છે. આ પ્રતિષ્ઠાપકો બે રીતે ઉપયોગમાં લેવામાં આવે છે :

- હોટ સ્પોટ ક્યાં આવેલા છે તે દર્શાવવા
- ક્લિક મારફતે ઉપયોગકર્તા દ્વારા પૂરાં પાડવામાં આવેલ પ્રતિષ્ઠાપકોને સરખાવવા.

ઉપયોગકર્તા બાજુ, બ્રાઉઝર ઉપયોગકર્તાની ક્લિકના પ્રતિષ્ઠાપકોને ઓળખી કાઢે છે અને ઉપયોગકર્તાને દર્શાવેલ વેબપેઈજ પર લઈ જાય છે. આ માટે Image ટેગમાં usemap ગુણધર્મ સાથે <map> અને <area> ટેગ ઉપયોગમાં લેવાય છે. map ટેગને name નામનો એક ગુણધર્મ હોય છે. name ગુણધર્મની કિંમત usemap ગુણધર્મની કિંમત સાથે મળવી જોઈએ. નીચેનું ઉદાહરણ જુઓ:

```

```

અહીં "#roadmap" એ આપણે બનાવેલા usemapના આઈડેન્ટિફાયરનું નામ છે. <map> અને </map> ટેગની વચ્ચે હોટ સ્પોટ પ્રતિષ્ઠાપકો અને તેમનું સંબંધિત જોડાણ (લિન્ક) દર્શાવવાની હોય છે. અહીં મંદિરના પ્રતિષ્ઠાપકો (518, 378, 70) છે અને તે વર્તુળાકાર છે. આપણે આ માહિતીને map ટેગમાં નીચે મુજબ આવરી લઈ શકીએ.

```
<map name="roadmap">
```

```
<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">
```

```
<!-- About co-ordinates of other hot spots ----- -->
```

```
</map>
```

ઉપરોક્ત HTML વિભાગની બીજી લીટીમાં આપેલ કો-ઓર્ડિનેટ્સ આકૃતિ 3.7માં દર્શાવેલ city.bmp ફાઈલનું મંદિર રજૂ કરે છે. મંદિર વર્તુળાકાર દર્શાવાયું હોવાને લીધે shapeની કિંમત આપણે "circle" આપી છે. ધારો કે, મંદિર લંબચોરસ આકારમાં દર્શાવવાનું હોત તો shape ની કિંમત તરીકે "rect" લખવું પડે. area ટેગ દ્વારા સ્થાનનો આકાર

દર્શાવાય તે જરૂરી છે. circle, rectangle અને polygon એ માન્ય આકારો છે. લંબચોરસ rect દ્વારા; વર્તુળ circle દ્વારા અને બહુકોણ poly દ્વારા દર્શાવાય છે. એના બદલે પૂરું નામ જેમ કે, rectangle પણ ઉપયોગમાં લેવાય છે. જુદા જુદા આકારો સાથે area ટેગને વર્ણવતું નીચેનું ઉદાહરણ જુઓ.

- **<area shape="poly" coords="32,301,183,301,239,352,188,399,32,399" alt="Central Mall" href="CentralMall.html">**
- **<area shape="rect" coords="32,432,233,532" alt="Fire Brigade" href="FireBrigade.html">**
- **<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">**

આકૃતિ 3.7માં દર્શાવેલ શહેરના દૃશ્યનો ઉપયોગ કરીને ઇમેજ મેપ (Image map) તૈયાર કરવા માટેનો સંપૂર્ણ HTML કોડ, કોડ-લિસ્ટિંગ 3.6માં દર્શાવાયો છે.

```
<html>
<body>
<p>Click on the location presented on map to look in detail:</p>
<!-- ----- -->


<!-- ----- -->
<map name="roadmap">

<area shape="rect" coords="46,37,219,141" alt="Office" href="Office.html">

<area shape="rect" coords="407,38,632,142" alt="Police Station" href="PoliceStation.html">

<area shape="poly" coords="32,301,183,301,239,352,188,399,32,399" alt="Central Mall"
href="CentralMall.html">

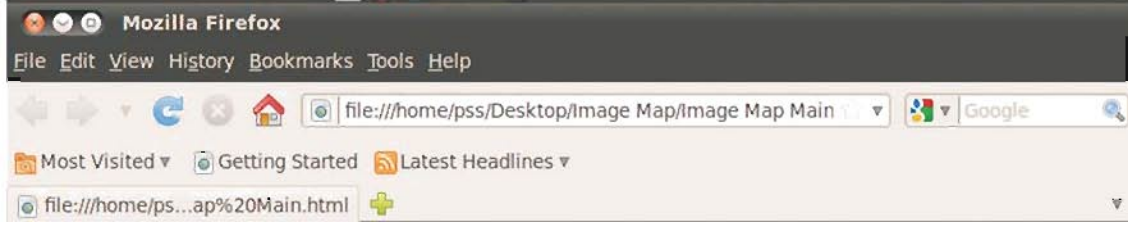
<area shape="rect" coords="32,432,233,532" alt="Fire Brigade" href="FireBrigade.html">

<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">

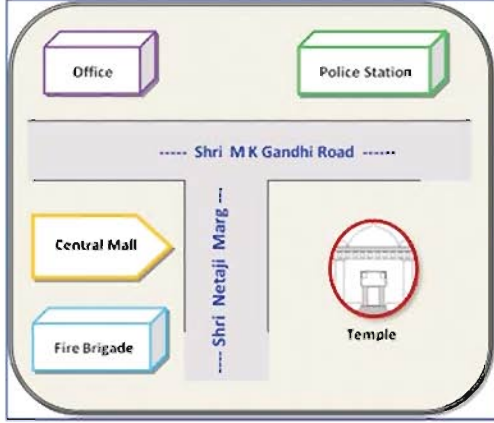
</map>
<!-- ----- -->
</body>
</html>
```

કોડ-લિસ્ટિંગ 3.6 : image map માટેના HTML કોડ

બ્રાઉઝરમાં કોડ લિસ્ટિંગ 3.6નું પરિણામ આકૃતિ 3.8માં દર્શાવ્યા મુજબનું દેખાશે :



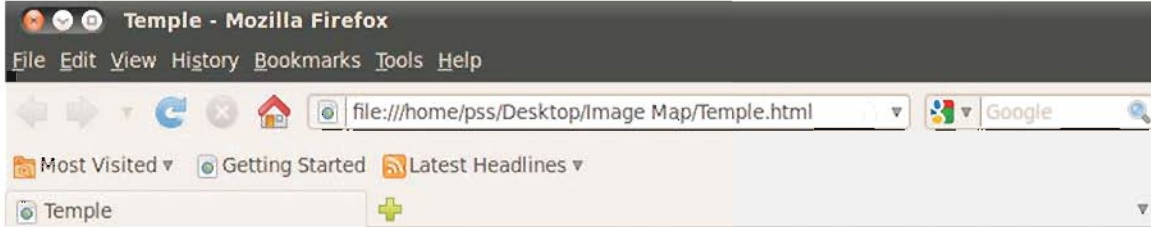
Click on the location presented on map to look in detail:



Done

આકૃતિ 3.8 : બ્રાઉઝરમાં ઇમેજ મેપ

જ્યારે તમે આકૃતિ 3.8માં દર્શાવેલ મંદિરની વર્તુળાકાર છબી પર ક્લિક કરશો ત્યારે તમને મંદિર વિશેની વિગતો દર્શાવતા નવા પાના પર લઈ જવામાં આવશે. મંદિરની વિગતો રજૂ કરતી અન્ય ફાઈલ દર્શાવતી આકૃતિ 3.9 જુઓ.



Our Big Temple

About the temple

The temple shown in the above Image is very famous "Sarv Dharma Mandir" located at the center of the city. The temple is first in its kind and consecrated various idols of God and Goddesses along with photographs of various religion and nature. People from all religion come to visit this temple from every corner of the world.

How to reach

The temple is 2 km. away from main bus stand and the railway station of the city. you may take city bus or auto to reach the temple. On foot it will take 15 minutes from the railway station.

Done

આકૃતિ 3.9 : મંદિરની વિગતો

આકૃતિ 3.9માં દર્શાવેલ વેબપેઈજ બનાવવા માટે જેની જરૂર પડે તે કોડ, કોડ-લિસ્ટિંગ 3.7 દર્શાવે છે :

```
<html>
<head> <title> Temple </title> </head>

<body>
  <center>
     </img>
  </center>

  <!-- ----- -->

  <h1> <font color="blue">About the temple </font> </h1>

  <p align="justify"> <b>

    The temple shown in the above image is very famous "Sarv Dharma Mandir"

    located at the center of the city. The temple is first in its kind and

    consecrated various idols of God and Goddesses along with photographs

    of various religion and nature. People from all religion come to visit

    this temple from every corner of the world.

  </b>
</p>
<!-- ----- -->
<h1> <font color="blue"> How to reach </font> </h1>

  <p align =justify> <b>

    The temple is 2 km. away from main bus stand and the railway station of the city.

    you may take city bus or auto to reach the temple. On foot it will take 15 minutes

    from the railway station.

  </b>
</p>
</body>
</html>
```

કોડ-લિસ્ટિંગ 3.7 : મંદિરની વિગતો માટેનો કોડ

એ જ રીતે, તમે કચેરી (Office.html), પોલિસમથક (PoliceStation.html), અગ્નિશામક દળ (FireBrigade.html) અને સેન્ટ્રલ મોલ (CentralMall.html) માટે HTML ફાઈલો તૈયાર કરી શકો. બધી જ ફાઈલો સ્વતંત્ર રીતે કામ કરે છે તે ચકાસી લો અને આખા શહેરનું સમગ્રતાયા ચિત્ર દર્શાવતી મુખ્ય HTML ફાઈલમાંથી ક્લિક કરીને તેમનો પ્રયોગ કરી જુઓ.

મલ્ટિમીડિયા ફાઈલોને જોડવી

HTML દસ્તાવેજમાં એન્કર ટેગનો ઉપયોગ કરીને નીચે દર્શાવ્યા પ્રમાણે વીડિયો તેમજ ઓડિયો ફાઈલને સંકલિત કરી શકાય છે:

** follow this recipe and enjoy delicious food...! **

આગળની લીટીમાં દર્શાવેલ ઉદાહરણમાં "food.mp4" નામની એક ચલચિત્ર (movie) ફાઈલ છે. તમારી પાસે પણ વીડિયો મુવી ફાઈલ તૈયાર હોવી જોઈએ. ઉપયોગકર્તા જ્યારે વેબપેઈજ પર દર્શાવાયેલ "Enjoy this... !" શબ્દો પર ક્લિક કરશે ત્યારે યોગ્ય સૉફ્ટવેરમાં વીડિયો ફાઈલ દર્શાવાશે. જો ઉલ્લેખ કરાયેલી મૂવી ફાઈલ અન્ય વેબસાઈટ કે અન્ય સ્થળે હશે તો સૌપ્રથમ તેને ઉપયોગકર્તાના કમ્પ્યુટરની ડિસ્કટરી પર કામચલાઉ ઇન્ટરનેટ ફાઈલ તરીકે સંગ્રહવામાં આવશે. મૂવી ફાઈલનો અમલ કરતી વખતે જો જરૂરી સૉફ્ટવેર નહીં મળે તો, બ્રાઉઝર તમને કમ્પ્યુટર અથવા ઇન્ટરનેટ પરથી વૈકલ્પિક સૉફ્ટવેર પસંદ કરવા પૂછશે.

HTML કોડમાં ધ્વનિ ફાઈલને ઉમેરવા માટેનું ઉદાહરણ નીચે આપેલ છે :

** Enjoy this song...! **

.mov ફાઈલ સ્વરૂપ ઉપરાંત વીડિયો ફાઈલ માટે અન્ય ફાઈલ સ્વરૂપો પણ છે. એમાંનાં થોડાં સ્વરૂપોમાં .avi, .wmv, .mpg અથવા .mpeg, અને .swf ગણી શકાય. આમાંના કેટલાંકને ફાઈલનો અમલ કરવા માટે વીડિયો પ્લેયર ડાઉનલોડ કરવાની જરૂર પડે છે.

સારાંશ

આ પ્રકરણમાં આપણે HTML દસ્તાવેજમાં છબીને કેવી રીતે ઉમેરવી તે શીખ્યા. HTML દસ્તાવેજમાં એક કે વધુ છબી ઉમેરવા ઉપરાંત align, hspace અને vspace જેવા ગુણધર્મોનો ઉપયોગ કરીને છબીઓને સ્ક્રીન પર કેવી રીતે ગોઠવવી તે પણ આપણે શીખ્યા. છબીને હોટ સ્પોટ કેવી રીતે બનાવવી તે પણ આપણે જોયું. માત્ર એક આખી છબીને જ નહીં બલ્કે, છબીના વિવિધ ભાગો(જેમ કે ભૌગોલિક નકશો)ને image map લાક્ષણિકતાનો ઉપયોગ કરીને જુદા જુદા વેબપેઈજઓ સાથે જોડી શકાય તે પણ આપણે જાણ્યું. અંતમાં, HTML સાથે ધ્વનિ અને વીડિયો ફાઈલ કેવી રીતે ઉમેરવી તે વિશેની માહિતી પણ મેળવી. આ પ્રકરણમાં આપવામાં આવેલી માહિતીનો ઉપયોગ કરીને, વિદ્યાર્થી HTML દસ્તાવેજમાં માત્ર સાદું લખાણ અને આંકડાઓ જ રજૂ કરતાં શીખવાને બદલે તે વેબપેઈજમાં વીડિયો, એનિમેશન, છબીઓ અને વિવિધ ધ્વનિને સંમિલિત કરવા માટે પણ સક્ષમ બનશે.

સ્વાધ્યાય

1. HTML દસ્તાવેજમાં છબીઓને કેવી રીતે ઉમેરી શકાય તે યોગ્ય ઉદાહરણ આપીને સમજાવો.
2. HTMLના image ટેગના વિવિધ ગુણધર્મો પર ટૂંક નોંધ લખો.
3. હોટસ્પોટની વ્યાખ્યા આપો. HTMLમાં છબીને કેવી રીતે હોટસ્પોટ બનાવી તેનો ઉપયોગ કરી શકાય તે ચર્ચો.
4. Image map પર ટૂંક નોંધ લખો.
5. નીચેના પ્રશ્નો માટે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ આપો:

(1) HTML દસ્તાવેજમાં વીડિયો ફાઈલ ઉમેરવા નીચેના પૈકી કયો ટેગ ઉપયોગમાં લેવાય છે ?

(a) img

(b) image

(c) href

(d) ime

- (2) image એલિમેન્ટ લીટી તોડતા (line break કરતાં) નથી, આને નીચેનામાંથી કયા નામે ઓળખવામાં આવે છે ?
- (a) ઓનલાઈન ઇમેજ (b) ઇનલાઈન ઇમેજ
(c) આઉટલાઈન ઇમેજ (d) બ્લેન્ક ઇમેજ
- (3) નીચેનામાંથી કયું માન્ય છબી ફાઈલ સ્વરૂપ છે ?
- (a) Img (b) Move (c) Mp3 (d) Png
- (4) જ્યારે આપણે છબીના alt ગુણધર્મનો ઉપયોગ કરીશું ત્યારે નીચેનામાંથી શું આપવામાં આવશે ?
- (a) વૈકલ્પિક વર્ણન (b) Alt કી વ્યાખ્યાયિત થાય
(c) વૈકલ્પિક છબી (d) વૈકલ્પિક HTML જોડાણ
- (5) નીચેનામાંથી કયો ગુણધર્મ છબીની ઊંચાઈ અને પહોળાઈની કિંમત પિક્સેલમાં દર્શાવે છે ?
- (a) Img src (b) Height અને width
(c) H અને V (d) આપેલા કોઈ પણ
- (6) નીચેનામાંથી કયું માન્ય છબી સ્વરૂપ નથી ?
- (a) Imv (b) Png (c) Bmp (d) Gif
- (7) આખી છબીને જોડાણ (લિન્ક) તરીકે દર્શાવવા નીચેનામાંથી કયો ખ્યાલ ઉપયોગમાં લેવાય છે ?
- (a) હોટ સ્પોટ તરીકે છબી (b) હોટ ટેક્સ્ટ
(c) સક્રિય જોડાણ (એક્ટિવ લિન્ક) (d) આપેલા કોઈ પણ
- (8) નીચેનામાંથી Image mapના બે પ્રકારો કયા છે ?
- (a) દુકાનદાર બાજુ અને ઉપયોગકર્તા બાજુ (b) સર્વર બાજુ અને ક્લાયન્ટ બાજુ
(c) વેપારી બાજુ અને પૂરા પાડનાર (સપ્લાયર) બાજુ (d) આપેલા બધા
- (9) image map ઉમેરવા માટે નીચેનામાંથી કઈ ટેગ ઉપયોગમાં લેવાય છે ?
- (a) Image name (b) Htemp (c) Map (d) Alt

પ્રાયોગિક સ્વાધ્યાય

1. આ પ્રકરણના કોડ લિસ્ટિંગ 3.1માં ચર્ચેલ ઉદાહરણને ધ્યાને લો. તે ચોકલેટ, સૂકો મેવો અને આઈસ્ક્રીમની છબી છાપે છે. HTML કોડને એ રીતે આગળ વધારો કે જેથી જો તમે ચોકલેટની છબી પર ક્લિક કરો તો તે તમને ચોકલેટ વિશેની માહિતી ધરાવતા નવા HTML પાના પર લઈ જાય. તમે ચોકલેટની થોડી વધુ છબી, ચોકલેટ વિશેની કેટલીક હકીકતો, ચોકલેટનો ઇતિહાસ અને ઘરે ચોકલેટ બનાવવાની વિધિ (રેસીપી) પણ ઉમેરી શકો.

જો તમે સૂકા મેવાની છબી પર ક્લિક કરો. તો તે તમને સૂકા મેવા વિશેની વધુ માહિતી ધરાવતા HTML પાના પર લઈ જશે. એ જ રીતે, જો તમે આઈસ્ક્રીમની છબી પર ક્લિક કરો તો તે તમને વિવિધ આઈસ્ક્રીમ વર્ણવતા નવા HTML પાના પર લઈ જશે.

2. Index.html નામનું પાનું તૈયાર કરો. તમારી જાણીતી ત્રણ હસ્તીઓ(personalities)ની નાની છબી મૂકો. આ હસ્તી તમારી શાળાના શિક્ષક, લેખકો, તમારા મિત્રો, ખેલાડીઓ હોઈ શકે. આ ઉપરાંત ઉપરોક્ત ત્રણેય વ્યક્તિઓની છબીની મોટા કદની છબી સાથે તેમના વિશેની વધુ માહિતી ધરાવતા ત્રણ વેબપેઈજ પણ બનાવો. Index પાના પરની દરેક નાની છબીઓ પર એવી રીતે જોડાણ (લિન્ક) ગોઠવો કે જેથી પસંદ કરેલ વ્યક્તિની સંપૂર્ણ માહિતી ધરાવતા પાના પર લઈ જાય. ત્રણેય નાની છબી સાથે તમારે ત્રણ જુદાં જોડાણ (લિન્ક) કરવાં પડશે.
(Hint : છબીમાંથી જોડાણ કરવા, એલિમેન્ટ ઉમેરો, અને એલિમેન્ટના href ગુણધર્મમાં મોટી છબીનું જોડાણ મૂકો.)
3. તમારા કુટુંબનું વૃક્ષ (family tree) તૈયાર કરવા, ઉપરોક્ત પ્રથમ ઉદાહરણમાં ચર્ચા કરેલ અભિગમનો ઉપયોગ કરો. તમારા દાદા અને દાદીની એક કરતાં વધુ છબીઓ સાથેનું વેબપેઈજ તૈયાર કરો. અન્ય પાનાં પણ એ રીતે તૈયાર કરો કે જેથી તમે જ્યારે તમારા દાદાની છબી પર ક્લિક કરો, ત્યારે થોડી વધુ સંબંધિત છબીઓ સાથે નવું વેબપેઈજ દર્શાવાય.
4. આ પ્રકરણમાં ચર્ચા કર્યા મુજબ શહેરનું સ્થળ, સ્થિતિ, ચિત્ર રજૂ કરતો image map તૈયાર કરો.
5. ફોટો ગેલેરી ઉમેરીને પ્રકરણ 2માં તમે તૈયાર કરેલ સ્કૂલ વેબસાઈટ પ્રોજેક્ટ પૂરો કરો. મુખ્ય વેબપેઈજ પર તમે તમારી શાળાનો લોગો મૂકો.
6. કોઈ હયાત વીડિયો ફાઈલનો ઉપયોગ કરો અને તેને વેબપેઈજમાં સામેલ કરો. તેને વિકલ્પરૂપે તમે તમારા મોબાઈલ અથવા અન્ય કોઈ સાધન દ્વારા એક વીડિયો ફાઈલ તૈયાર કરી શકો અને તેને વેબપેઈજમાં સામેલ કરી શકો.





HTMLમાં યાદી અને કોષ્ટકનો ઉપયોગ

યાદીનું વ્યવસ્થાપન

માહિતી રજૂ કરતી વખતે, ઘણી વખત આપણે વિવિધ અલગ-અલગ મુદ્દાઓ પણ રજૂ કરવાની જરૂર પડતી હોય છે. આવા મુદ્દાઓને ફકરા સ્વરૂપે સળંગ લખવાને બદલે આવા જુદા જુદા મુદ્દા ક્રમ સંખ્યા સહિત કે તેના વગર યાદી સ્વરૂપે રજૂ કરવામાં આવે છે. મુદ્દાઓને આ રીતે રજૂ કરવાથી રજૂઆતની સ્પષ્ટતામાં વધારો થાય છે અને તે વાચકને વાંચવામાં અને સમજવામાં સરળતા બક્ષે છે.

તો ચાલો, એક ઉદાહરણ જોઈએ. જ્યારે તમે બજારમાંથી અમુક ચીજો ખરીદવા ઇચ્છતા હોવ ત્યારે તમારે તેને યાદ રાખવાની જરૂર પડે છે. આકૃતિ 4.1 આવી કેટલીક વસ્તુઓની સરસ રીતે યાદી દર્શાવે છે, જે ખરીદી કરવા જતી વખતે તમે સાથે રાખી શકો છો.

Items to be purchased

- Chocolates
- Notebook
- Practice book
- Ballpen
- Markers

આકૃતિ 4.1 : યાદીનું ઉદાહરણ

HTML યાદીના મૂળભૂત ત્રણ પ્રકાર છે: (1) ક્રમબદ્ધ યાદી (ordered list), (2) અક્રમિક યાદી (unordered list) અને (3) વર્ણનાત્મક યાદી (descriptive list). આ તમામ પ્રકાર આ પ્રકરણના આગળ આવતા વિભાગોમાં ચર્ચવામાં આવેલ છે.

અક્રમિક યાદી (Unordered List)

આકૃતિ 4.1માં દર્શાવ્યા મુજબ, અક્રમિક યાદીમાં કોઈ ચિહ્ન વડે યાદીના મુદ્દા દર્શાવાય છે. આ ચિહ્નને 'બુલેટ' (bullet) પણ કહે છે. આકૃતિ 4.1માં દર્શાવેલ યાદી એ બુલેટ તરીકે '●' સાથેની અક્રમિક યાદી છે. HTMLમાં અક્રમિક યાદી તૈયાર કરવા માટે અને ટેગની જોડી ઉપયોગમાં લેવાય છે. યાદીના એક પછી એક દરેક મુદ્દા અને ટેગની જોડીમાં આવરીને દર્શાવાય છે. અક્રમિક યાદી તૈયાર કરતાં HTML ઉદાહરણ કોડ-લિસ્ટિંગ 4.1માં દર્શાવેલ છે.

```
<html>
<!-- ----- -->
<body>
<font color="Blue">
    <h1>An Unordered List:</h1>
</font>
<!-- ----- -->
<ul>
<font size="6">
```

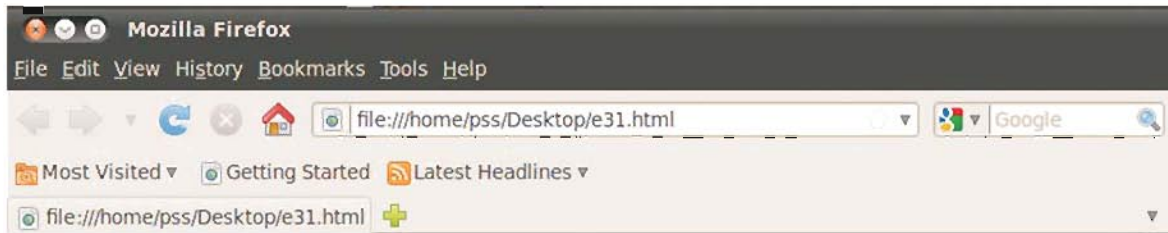
```

<b>
  <li>Chocolates</li>
  <li>Notebook</li>
  <li>Practice book</li>
  <li> Ball pen </li>
  <li>Markers </li>
</b>
</font>
</ul>
<!-- ----->
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.1 : અકમિક યાદી

SciTE એડિટરમાં કોડ-લિસ્ટિંગ 4.1માં દર્શાવેલ કોડ ટાઇપ કરીને બ્રાઉઝરમાં તેનું પરિણામ ચકાસો. તેનું પરિણામ આબેહૂબ આકૃતિ 4.2માં દર્શાવ્યા મુજબ દેખાશે.



An Unordered List:

- . Chocolates
- . Notebook
- . Practice book
- . Ball pen
- . Markers

Done

આકૃતિ 4.2 : બ્રાઉઝરમાં અકમિક યાદી

બુલેટ બદલવી

સામાન્ય રીતે ભરેલા ગોળ ચિહ્નને બુલેટ તરીકે લેવાય છે. પરંતુ જો આપણે એ ચિહ્ન બદલવું હોય તો, તમારે type નામના ગુણધર્મનો ઉપયોગ કરવો પડે. બુલેટના વિવિધ પ્રકારોમાં ભરેલ વર્તુળ, ખાલી વર્તુળ (disc) અને ચોરસ હોઈ શકે. નીચેનું ઉદાહરણ જુઓ:

```
<ul type="square">
```

બુલેટ તરીકે ખાલી ગોળ ચિહ્નનો ઉપયોગ કરવા માટે આપણે નીચે મુજબ ટેગ લખી શકીએ :

<ul type="circle">

કોડ-લિસ્ટિંગ 4.1માં આપેલા HTML કોડને બદલીને વારાફરતી જુદી જુદી જાતની બુલેટનું અવલોકન કરો.

બુલેટ વગરની યાદી

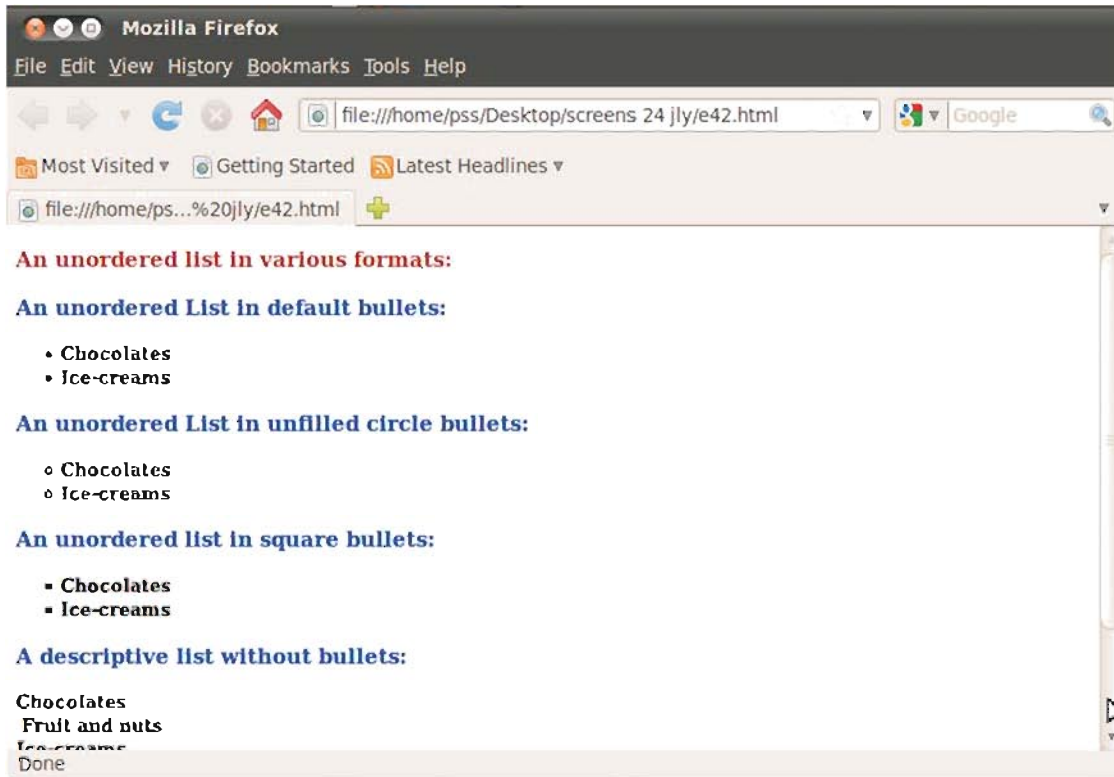
જો કોઈ કિસ્સામાં તમારે યાદીમાં મુદ્દાની આગળ કોઈ પણ જાતની બુલેટ ન દર્શાવાય તેવું ઇચ્છતા હોય તો, તમે <dl> અને </dl> નો ઉપયોગ કરીને વર્ણનાત્મક યાદી તૈયાર કરી શકો. અહીં, 'dl' એ description listનું ટૂંકું નામ છે. વર્ણનાત્મક યાદીની અંદર વિવિધ મુદ્દા રજૂ કરવા આપણે દરેક નવા મુદ્દાને <dt> અને </dt> ટેગનો ઉપયોગ કરવો પડે. એટલે કે, મુદ્દા તરીકે Chocolate દર્શાવવા માટે નીચે મુજબ ટેગ લખી શકાય:

<dt>

<dt> Chocolates </dt>

</dl>

એમાંય વળી, પેટાવિગત દર્શાવવા માટે આપણે વર્ણનાત્મક ટેગ <dd> અને </dd> નામની જોડીનો ઉપયોગ કરી શકીએ. એટલે કે, આપણે ચોકલેટ અને આઈસ્ક્રીમની વિવિધ જાતો ઉમેરી શકીએ. આકૃતિ 4.3 વેબપેઈજમાં જુદી જુદી જાતની બુલેટનો ઉપયોગ દર્શાવે છે.



આકૃતિ 4.3 : બુલેટના જુદા જુદા પ્રકારો

આકૃતિ 4.3 જેવું પરિણામ દર્શાવવા માટેના HTML કોડ કોડ-લિસ્ટિંગ 4.2માં દર્શાવેલ છે.

```
<html>
<body>
<font color ="Brown">
    <h1>An Unordered List in various formats: </h1>
</font>
<!-- ----->
```



```

<font color ="Blue">   <h1>An unordered List in default bullets: </h1>   </font>
<ul>
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
</b>
</font>
</ul>
<!-- ----- -->
<font color ="Blue">   <h1>An unordered List in unfilled circle bullets: </h1>   </font>
<ul type="circle">
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
</b>
</font>
</ul>
<!-- ----- -->
<font color ="Blue">   <h1>An unordered list in square bullets: </h1> </font>
<ul type="square">
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
    <li>Dry fruits    </li>
</b>
</font>
</ul>
<!-- ----- -->
<font color ="Blue">
    <h1>A descriptive list without bullets: </h1>
</font>
<dl>
<font size="6">
<b>
    <dt>Chocolates    </dt>
        <dd> Dark </dd>
        <dd> Fruit and nuts </dd>
        <dd> Milk </dd>

```

```

<dt>Ice-creams </dt>
<dd> Vanilla </dd>
<dd> Chocolate chips </dd>
<dt>Dry fruits </dt>
<dd> Almonds </dd>
<dd> Cashew nuts </dd>
</b>
</font>
</dl>
<!-- ----- -->
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.2 : વિવિધ પ્રકારની બુલેટ્સ માટેના HTML કોડ

યાદીનું મથાળું

યાદીનું મથાળું આપવા માટે આપણે ફક્ત <lh> ટેગનો ઉપયોગ કરવો પડે. સ્વાભાવિક રીતે જ, મથાળું યાદીની પહેલાં જ આવવું જોઈએ. આમ, અને ટેગની જોડી લખતાં પહેલાં જ આપણે જરૂરી મથાળા સાથે <lh> ટેગનો ઉપયોગ કરવો પડે. નીચે આપેલ ઉદાહરણ જુઓ :

<lh> My Shopping List </lh>

ક્રમબદ્ધ યાદી

ક્રમબદ્ધ યાદીમાં બુલેટ્સને બદલે ક્રમ દર્શાવતા આંકડા કે અક્ષરો ધરાવતા મુદ્દાઓ હોય છે. HTML માં આવી ક્રમબદ્ધ યાદી બનાવવા માટે અને ટેગનો ઉપયોગ થાય છે. હરહંમેશની જેમ યાદીમાંના દરેક મુદ્દાને અને ટેગની વચ્ચે આવરીને દર્શાવાય છે. ક્રમબદ્ધ યાદી બનાવતું HTML ઉદાહરણ કોડ-લિસ્ટિંગ 4.3માં દર્શાવેલ છે.

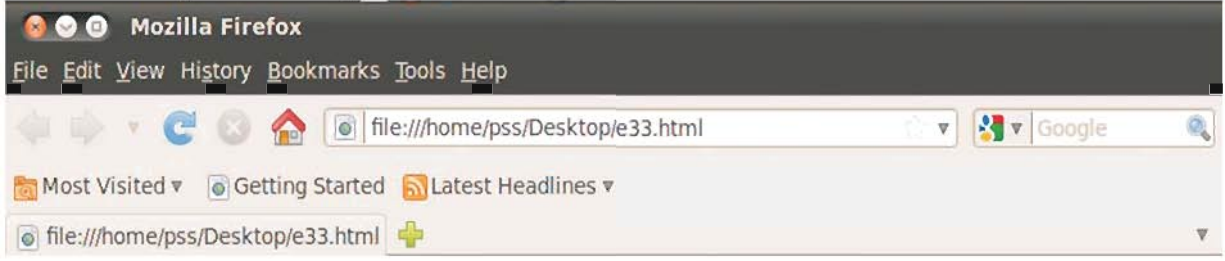
```

<html>
<body>
<font color ="Blue">
<h1>An ordered list: </h1>
</font>
<!-- ----- -->
<ol >
<font size="6">
<b>
<li>Chocolates </li>
<li>Ice-creams </li>
<li>Dry fruits </li>
</b>
</font>
</ol>
<!-- ----- -->
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.3 : ક્રમબદ્ધ યાદી માટે HTML કોડ

કોડ-લિસ્ટિંગ 4.3નું પરિણામ આકૃતિ 4.4માં દર્શાવેલ છે.



An ordered list:

1. Chocolates
2. Ice-creams
3. Dry fruits

Done

આકૃતિ 4.4 : ક્રમબદ્ધ યાદી

સામાન્ય રીતે, ક્રમબદ્ધ યાદી હંમેશાં ક્રમ 1થી શરૂ થાય છે. તમે જો ઇચ્છો તો કોઈ ચોક્કસ ક્રમ સંખ્યાથી પણ યાદી શરૂ કરી શકો. ધારો કે, તમારે યાદીના પહેલાં મુદ્દાની શરૂઆત ક્રમ 6થી કરવી હોય તો તમે નીચે દર્શાવ્યા મુજબ `` ટેગ સાથે `start` ગુણધર્મનો ઉપયોગ કરી શકો.

`<ol start = "6">`

ક્રમ 6થી યાદીની શરૂઆત કરવા માટે કોડ-લિસ્ટિંગ 4.3માં આપેલ HTML કોડને સુધારો.

અક્ષરક્રમ ધરાવતી ક્રમબદ્ધ યાદી

આંકડાની જગ્યાએ, આપણે ક્રમ દર્શાવવા અંગ્રેજી મૂળાક્ષરો 'A', 'B', 'C' અથવા 'a', 'b', 'c'. આપણે રોમન અંકોનો પણ ઉપયોગ કરી શકીએ. આવું `` ટેગની સાથે `type` ગુણધર્મનો ઉપયોગ કરીને કરી શકાય. ટેબલ 4.1 type ગુણધર્મ માટેની શક્ય ક્રિમત રજૂ કરે છે.

ક્રિમત	વર્ણન
1	અંક
A	અંગ્રેજી મોટા મૂળાક્ષરો
a	અંગ્રેજી નાના મૂળાક્ષરો
I	રોમન મોટા અંકો
i	રોમન નાના અંકો

કોષ્ટક 4.1 : Type ગુણધર્મની ક્રિમતો

કોડ-લિસ્ટિંગ 4.4માં HTML કોડ આપવામાં આવેલ છે, આ કોડ એક અંગ્રેજી મૂળાક્ષર સાથેની અને એક ક્રમ 10થી શરૂ થતા રોમન અંક સાથેની એમ બે ક્રમબદ્ધ યાદીઓ રજૂ કરશે.

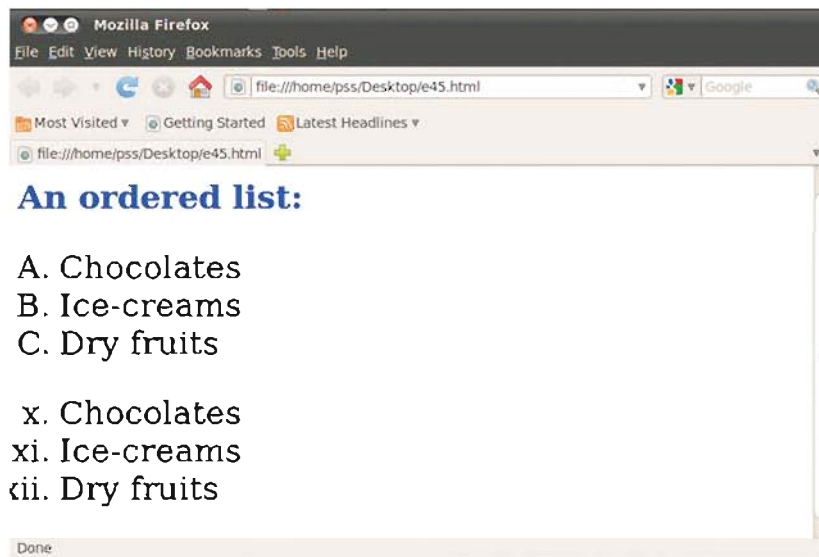
```

<html>
<body>
<font color="Blue">
  <h1>An ordered list: </h1>
</font>
<!-- ----->
<font size="6">
<b>
  <ol type=A>
    <li>Chocolates </li>
    <li>Ice-creams </li>
    <li>Dry fruits </li>
  </ol>
</b>
</font>
<!-- ----->
<font size="6">
<b>
  <ol type='i' start="10">
    <li>Chocolates </li>
    <li>Ice-creams </li>
    <li>Dry fruits </li>
  </ol>
</b>
</font>
<!-- ----->
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.4 : બુલેટ તરીકે રોમન અંકો છાપવા માટેના HTML કોડ

HTML કોડ 4.4નું પરિણામ આકૃતિ 4.5માં આપવામાં આવેલ છે. આ રીતે તમે type ગુણધર્મની જુદી જુદી કિંમતો સાથે પ્રયત્ન કરી શકો છો.



આકૃતિ 4.5 : અંગ્રેજી મૂળાક્ષરો અને રોમન અંકો સાથેની ક્રમબદ્ધ યાદી

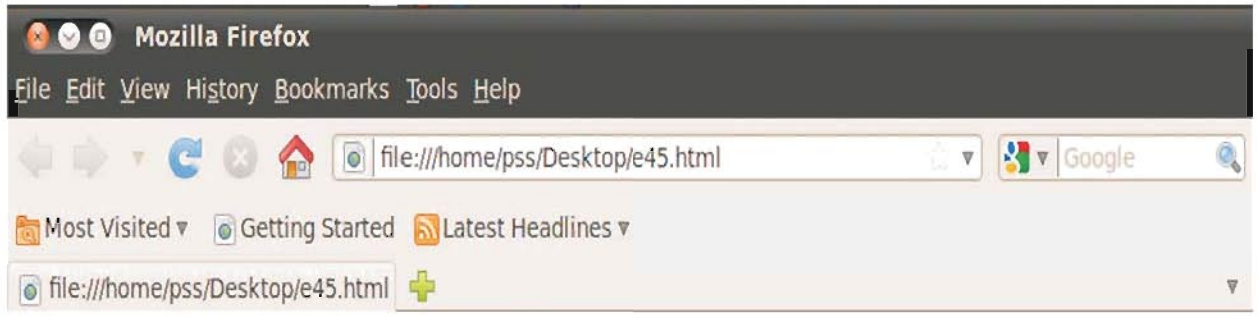
નેસ્ટેડ યાદી

કોઈ એક યાદીમાંય વળી બીજી યાદી સમાવવી પડે તેવી પણ જરૂરિયાત હોય છે. આને યાદીનું નેસ્ટિંગ અથવા નેસ્ટેડ યાદી કહે છે. વળી, આવા કિસ્સામાં પ્રથમ યાદી અને એની અંદર રહેલી બીજી યાદીનો પ્રકાર જુદો જુદો પણ હોઈ શકે. ઉદાહરણરૂપે HTML કોડ એ કોડ-લિસ્ટિંગ 4.5માં આપેલ છે.

```
<html>
<body>
<font color ="Blue">
    <h1>An ordered list: </h1>
</font>
<!-- ----- -->
<ul >
<font size= "6 ">
<b>
    <li>Chocolates    </li>
    <font size= "4">
    <ol type="a">
        <li> Dark chocolates    </li>
        <li> Fruit and nuts Dark    </li>
    </ol>
    </font>
    <!-- ----- -->
    <li>Ice-creams    </li>
    <ol type="a">
        <font size="4">
            <li> Venilla    </li>
            <li> Chocolate chips    </li>
        </font>
    </ol>
    <!-- ----- -->
</b>
</font>
</ul>
</body>
</html>
```

કોડ-લિસ્ટિંગ 4.5 : નેસ્ટેડ યાદી માટેના HTML કોડ

કોડ-લિસ્ટિંગ 4.5નું પરિણામ આકૃતિ 4.6માં દર્શાવેલ છે.



An ordered list:

. Chocolates

- a. Dark chocolates
- b. Fruit and nuts Dark

. Ice-creams

- a. Venilla
- b. Chocolate chips

Done

આકૃતિ 4.6 : HTMLમાં નેસ્ટેડ યાદી

લખાણને વધુ માળખાગત રીતે રજૂ કરી શકે તેવાં માળખાં પણ છે. આવા એક માળખાને કોષ્ટક કહે છે. નીચેનો વિભાગ HTMLમાં કોષ્ટક રજૂ કરે છે.

HTMLમાં કોષ્ટકનું વ્યવસ્થાપન

માહિતીની યોગ્ય સ્વરૂપે રજૂઆત સમજવામાં અને ઉપયોગ કરવામાં સરળતા બક્ષે છે. શું તમે કોઈ ખરીદીના બિલને સળંગ ફકરા સ્વરૂપે છપાવેલું હોય તેવી કલ્પના કરી શકો છો. વિવિધ વિષયોના ગુણ રજૂ કરતા તમારા ગુણપત્રક(માર્કશીટ)નું શું ? તે આકૃતિ 4.7માં દર્શાવ્યા જેવી મળે તો તે તમને વાંચવા માટે ચોક્કસપણે અઘરી પડે. તે માત્ર વધુ પ્રયત્ન માગી લે તેવી જ નહીં બલ્કે વધુ સમય લગાડનારી પણ હશે. ઉપરાંત, તેના દ્વારા માહિતીનું ખોટું અર્થઘટન પણ થઈ જાય એવું બને.

Modern School Affiliated to Star Education, Gujarat Mr. Arvind B Patel Seat No 123

March 2013 Subjects: English, Gujarati, Science, Maths, SS*

Internal Marks (40): 20, 30, 25, 35, 25 External Marks (60): 35, 32, 48, 40, 35

Class: First Class SS*: Social Science

આકૃતિ 4.7 : વિચિત્ર પ્રકારના ગુણપત્રકનું ઉદાહરણ

હવે, આકૃતિ 4.8 જુઓ. દેખીતી રીતે જ તેમાં કોઈ ગૂંચવણ નથી. શાળાનું નામ, વિદ્યાર્થીનું નામ, તારીખ, તમામ વિષયોના ગુણ અને પરિણામ સ્પષ્ટપણે દેખાઈ આવે છે.

Modern School*Affiliated to Star Education, Gujarat***Mr. Arvind B Patel****Seat No 123****March 2013**

Subject:	English	Gujarati	Science	Maths	SS *
Internal Marks (40):	20	30	25	35	25
External Marks (60):	35	32	48	40	35

Class: First class**SS*: Social Science****આકૃતિ 4.8 : યોગ્ય સ્વરૂપના ગુણપત્રકનો નમૂનો**

માહિતીની સુનિયોજિત રજૂઆત માટે HTMLમાં કોષ્ટક વધુ કાર્યક્ષમ છે. કોષ્ટકમાં ખાનાંઓ સ્વરૂપે માહિતી રજૂ થાય છે. તમારા ગણિતની ગણતરીઓ માટે કદાચ તમે ઊભાં અને આડાં ખાનાં ધરાવતી વિશિષ્ટ નોંધપોથી વાપરી હશે. એ જ રીતે, કોષ્ટકમાં માહિતીને ઊભી અને આડી હાર સ્વરૂપે રજૂ કરી શકાય. હવે પછીના વિભાગમાં HTML ટેગના ઉપયોગ દ્વારા વેબપેઈજ પર કેવી રીતે એક નાનું કોષ્ટક દર્શાવી શકાય. તે વર્ણવવામાં આવ્યું છે. એ પછી, કોષ્ટકનું સર્જન કરતાં દરેક ટેગને વિગતવાર જોઈશું.

એક નાના કોષ્ટકનું સર્જન

HTMLમાં કોષ્ટક તૈયાર કરવા <table> અને </table> ટેગનો ઉપયોગ થાય છે. એટલે કે, કોષ્ટકની શરૂઆત અને અંત એ આ બે ટેગ દ્વારા દર્શાવાય છે. <table> અને </table> ટેગની વચ્ચે આપણે કોષ્ટકનું શીર્ષક, કોષ્ટકની કિનારી, કોષ્ટક માટે ઊભી અને આડી હરોળ જેવા ગુણધર્મોનો ઉપયોગ કરી શકીએ.

કોષ્ટકનું મુખ્ય લખાણ ઊભી હાર અને આડી હરોળ સ્વરૂપે ગોઠવાય છે. કોષ્ટકમાં આડી હરોળ <tr> ટેગ દ્વારા દર્શાવવામાં આવે છે. કોષ્ટકની પ્રથમ આડી હરોળ મથાળા માટેની હોય છે, જે <th> ટેગ દ્વારા દર્શાવાય છે. એ પછીની બાકીની આડી હરોળ <td> ટેગ દ્વારા દાખલ કરવામાં આવે છે. કોડ-લિસ્ટિંગ 4.6 જુઓ, જેમાં ઉદાહરણરૂપ HTML કોડ દર્શાવવામાં આવેલ છે.

```

<html>
<body>
<h1>My First Table </h1>
<p>My first table is as follows.</p>
<table border="2">
<tr>
<td> This is Row 1, Column 1 </td>
<td> This is Row 1, Column 2 </td>
</tr>
<tr>
<td> This is Row 2, Column 1 </td>
<td> This is Row 2, Column 2 </td>
</tr>
</table>
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.6 : સાદા ટેબલ માટેના HTML કોડ

તમે તમારા એડિટરમાં આ કોડ લખીને બ્રાઉઝરમાં પરિણામ જોઈ શકો. બ્રાઉઝરમાં જોવાથી પરિણામ આકૃતિ 4.9 જેવું જોવા મળશે.



આકૃતિ 4.9 : વેબપેઈજ પર દર્શાવાયેલ એક સાદું HTML કોષ્ટક

કોડ-લિસ્ટિંગ 4.6માં દર્શાવેલ HTML કોડ થોડા ગુણધર્મો અને ભાગ સાથે table ટેગનો ઉપયોગ કરે છે. Table ટેગની અંદર ઉપયોગમાં લેવાયેલ પ્રથમ ગુણધર્મ પોતે જ 2 પિક્સેલ કદની કિનારી દર્શાવવા માટેનો border ગુણધર્મ છે. કોષ્ટકમાં એ પછી <tr> ટેગ સાથે આડી હરોળ વ્યાખ્યાયિત કરે છે. આકૃતિ 4.9માં દર્શાવેલ કોષ્ટકને બે આડી હરોળ છે, આ માટે <tr> ટેગની બે જોડી ઉપયોગમાં લેવી પડે.

આડી હરોળની અંદર <td> ટેગની મદદથી ઊભી હરોળ વ્યાખ્યાયિત કરી શકાય છે. આ ઉદાહરણમાં ઉપયોગમાં લેવાયેલ ટેગ અને ગુણધર્મોની ચર્ચા આ પ્રકરણમાં પછીથી કરવામાં આવેલ છે. ઉદાહરણનો ઉદ્દેશ કોષ્ટકની રચનામાં ઊભી અને આડી હરોળ સમજવા માટે છે. આ પ્રકારની વ્યવસ્થાને ખાનાવાળી માહિતી (grid type) તરીકે પણ ઓળખવામાં આવે છે. તો ચાલો, કોષ્ટક માટેના અન્ય ટેગ વિશે વિસ્તૃત માહિતી મેળવીએ.

Table ટેગ અને તેના ગુણધર્મો

કોષ્ટકને વ્યાખ્યાયિત કરવા માટે ઉપયોગમાં લેવાતો પ્રથમ ટેગ એ table ટેગ છે. <table> ટેગમાં ઘણા ગુણધર્મો સમાવી શકાય છે. તેમાંના કેટલાક આજકાલ ખાસ પ્રચલિત નથી. જો કે, તેમ છતાં તમે એને વાપરી શકો. આવા ગુણધર્મો કોષ્ટક 4.2માં દર્શાવાયેલ છે.

align	આ ગુણધર્મ કોષ્ટકની ગોઠવણી માટે ઉપયોગી છે.
bgcolor	આ ગુણધર્મ કોષ્ટકના બેકગ્રાઉન્ડ રંગ બાબતે જણાવે છે.
border	આ ગુણધર્મ કોષ્ટકની કિનારી બાબતે જણાવે છે.
cellpadding	ખાનાની ધાર (edges of cell) અને ખાનાની અંદરની માહિતી વચ્ચે નક્કી કર્યા મુજબની જગ્યા છોડવા આ ગુણધર્મ દ્વારા જણાવી શકાય છે.
cellspacing	આ ગુણધર્મ કોષ્ટકના દરેક ખાનાની વચ્ચે કેટલી જગ્યા રાખવી તે જણાવે છે.
dir	આ ગુણધર્મ કોષ્ટકમાં દર્શાવાતા લખાણની દિશા નક્કી કરવા માટે ઉપયોગી છે.
frame	આ ગુણધર્મ કોષ્ટકની છેક બહારની બાજુની કિનારીને નિયંત્રિત કરે છે.
rules	આ ગુણધર્મ કોષ્ટકની અંદરની કિનારીની રજૂઆતને નિયંત્રિત કરે છે.
summary	આ ગુણધર્મ કોષ્ટકનું વર્ણન રજૂ કરે છે.
width	આ ગુણધર્મ કોષ્ટકની પહોળાઈ કેટલી રાખવી તે જણાવે છે.

કોષ્ટક 4.2 : Table ટેગના ગુણધર્મો

સામાન્યપણે ઉપયોગમાં લેવાતા કેટલાક ગુણધર્મો નીચે ચર્ચવામાં આવેલ છે :

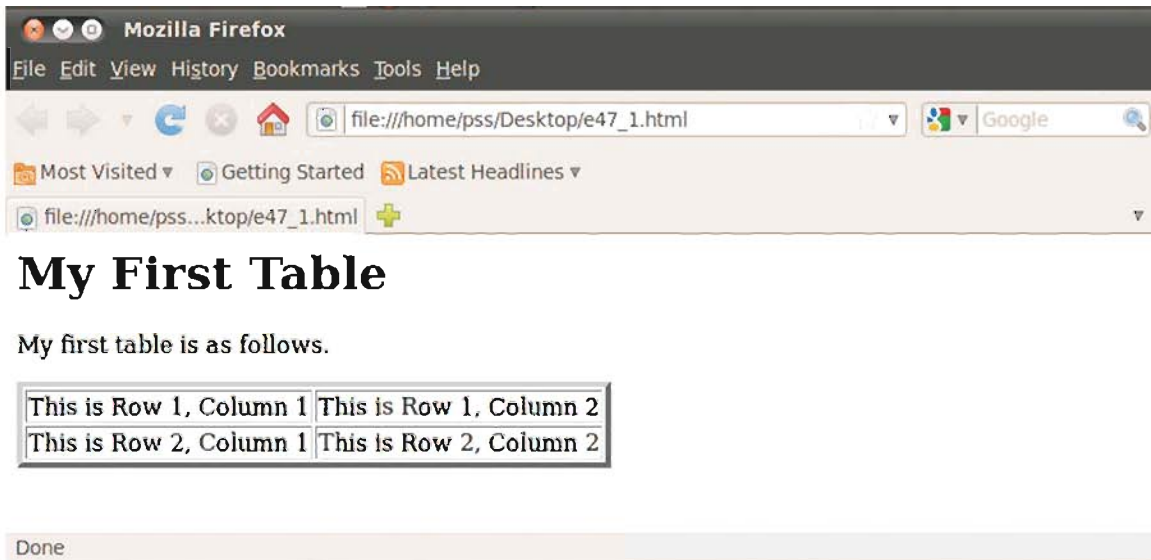
Border ગુણધર્મ

કોડ-લિસ્ટિંગ 4.6માં ચર્ચેલ ઉદાહરણમાં border ગુણધર્મનો ઉપયોગ કરીને 2 પિક્સેલ માપની કિનારી બનાવવામાં આવે છે. આ ગુણધર્મ કોષ્ટકની ફરતે ચારે બાજુ તેમજ દરેક ખાનાની આસપાસ કિનારી દોરે છે. કિનારીની પહોળાઈ પિક્સેલ સ્વરૂપે અપાય છે. Border ગુણધર્મનો ઉપયોગ કરવો મરજિયાત છે. જો તમે ગુણધર્મ તરીકે 0 કિંમત આપશો તો કોઈ કિનારી દેખાશે નહીં. અગાઉ કોડ-લિસ્ટિંગ 4.6નો ઉપયોગ કરીને તૈયાર કરાયેલ HTML કોડ યાદ કરો. ચાલો, નીચે દર્શાવ્યા પ્રમાણે આપણે Border ગુણધર્મની કિંમત સુધારીએ :

```
<table border="4" align="left">
```

હવે, ફરી એકવાર કોષ્ટકનો દેખાવ ચકાસી લો. તે આકૃતિ 4.10માં દર્શાવ્યા મુજબ દેખાવી જોઈએ.

આગળના ફકરામાં આપેલ ઉદાહરણમાં આપણે ઉપયોગમાં લીધેલ એક વધુ ગુણધર્મ કદાચ તમારા ધ્યાને આવ્યો હશે. આ ગુણધર્મ છે align. શું તમે કોઈ બાબતની નોંધ લીધી ? તમે કદાચ કોઈ ફેરફાર નહીં નોંધ્યો હોય. કોષ્ટક જમણી બાજુ ગોઠવવા ગોઠવણી (alignment) બદલો. હવે તમે કોષ્ટક ગોઠવણી(table alignment)માં ફેરફાર જોઈ શકશો. અન્ય ગોઠવણી (alignment) ની કિંમતો પછીથી જોઈશું.



આકૃતિ 4.10 : કિનારીની બદલાયેલ કિંમત સાથેનાં કોષ્ટક

Bgcolor ગુણધર્મ

ચાલો, કોષ્ટકમાં બેકગ્રાઉન્ડ રંગ મૂકીને આપણે કોષ્ટકને વધુ રંગીન અને આકર્ષક બનાવીએ. bgcolor ગુણધર્મ કોષ્ટકનો બેકગ્રાઉન્ડ રંગ ગોઠવવાનું કાર્ય કરે છે. આ ગુણધર્મની કિંમત કોઈ રંગનું નામ અથવા છ આંકડાનો સોળઅંકી સંખ્યા લેખન પદ્ધતિનો કોડ હોઈ શકે. તાજેતરમાં તમે જેનું પરીક્ષણ કરતા હતા તે HTML કોડ (કોડ-લિસ્ટિંગ 4.6માં આપ્યા મુજબ)ની પ્રથમ લીટી સુધારો.

```
<table border="4" align="right" bgcolor="yellow">
```

બ્રાઉઝરમાં કોડ નિહાળીને પરિણામ ચકાસો.

Cellpadding ગુણધર્મ

જો કોષ્ટકના બે ખાનાં એકબીજાની બાજુમાં મૂકવામાં આવે અને બંનેમાં શાબ્દિક લખાણ હોય તો થોડી સમસ્યા ઊભી થઈ શકે. જો બંને ખાનાંની ધાર અને લખાણ વચ્ચે પૂરતી જગ્યા નહીં હોય તો બંનેના શબ્દો ભેગા થઈ

ગયેલા લાગશે અને તેના કારણે વાંચવામાં તકલીફ પડશે. એ જ રીતે, જો દરેક ખાનાની ફરતે કિનારી હશે અને લખાણ આ કિનારીને અડતું હશે, તો તે વાંચવામાં તકલીફ પડશે. આવા કિસ્સામાં જો ખાનાઓમાં થોડી વધુ જગ્યા (space as padding) રાખવામાં આવે તો તે ખાનાઓનાં લખાણ વાંચવામાં સરળતા રહે.

કોષ્ટકના બે ખાનાઓની કિનારી અને તેના લખાણની વચ્ચે થોડી જગ્યા મૂકવા cellpadding ગુણધર્મનો ઉપયોગ કરવામાં આવે છે. cellpadding ગુણધર્મ કોષ્ટકના ખાનાઓની દરેક દીવાલ(wall of cell)ની અંદરની બાજુ થોડી જગ્યાની ગાદી (pad) બનાવે છે. આ ગુણધર્મની કિંમત ટકાવારીમાં અથવા પિક્સેલમાં આપી શકાય. નીચેનું ઉદાહરણ જુઓ.

```
<table border="4" align="center" bgcolor="pink" cellpadding="25">
```

આના બદલે નીચે મુજબ તમે ટકાવારીનો ઉપયોગ કરી શકો :

```
<table border="4" align="center" bgcolor="pink" cellpadding="20%">
```

તમે કદાચ એ નોંધ્યું હશે કે, આપણે align અને bgcolor ગુણધર્મની કિંમતો પણ બદલી છે. કોષ્ટકના બદલાયેલા દેખાવને બ્રાઉઝરમાં ચકાસો.

Cellspacing ગુણધર્મ

cellspacing ગુણધર્મ, કોષ્ટકના ખાનાઓની વચ્ચે જગ્યા કરવા માટે ઉપયોગમાં લેવાય છે. કેટલી જગ્યા રાખવી તે માટેની કિંમત ટકાવારીમાં અથવા પિક્સેલમાં દર્શાવી શકાય છે. અહીં, ટકાવારી એટલે કોષ્ટકની દરેક ખાનાની પહોળાઈના ટકા. નીચેનું ઉદાહરણ જુઓ :

```
<table border="4" align="center" bgcolor="yellow" cellspacing="20%">
```

અથવા

```
<table border="4" align="center" bgcolor="yellow" cellspacing="25">
```

Width ગુણધર્મ

width ગુણધર્મ કોષ્ટકની પહોળાઈ નક્કી કરવા માટે ઉપયોગમાં લેવાય છે. આ પહોળાઈ કેટલી રાખવી તેની કિંમત હરહંમેશની જેમ પિક્સેલમાં અથવા ઉપલબ્ધ જગ્યાની ટકાવારીમાં આપી શકાય છે.

```
<table border="4" align="center" bgcolor="pink" cellspacing="25" frame="box" rules="cols" width="50%">
```

અથવા

```
<table border="4" align="center" bgcolor="pink" cellspacing="25" frame="box" rules="cols" width="50">
```

Align ગુણધર્મ

align ગુણધર્મ આડી હારમાં આવેલા તમામ ખાનાઓના લખાણનું સ્થાન નક્કી કરે છે. સાદા લખાણની જેમ જ કોષ્ટકના ખાનાની અંદર લખાણ પણ ગોઠવી શકાય છે. કોષ્ટક 4.3 align ગુણધર્મ માટેની શક્ય કિંમતોની યાદી રજૂ કરે છે.

કિંમત	વર્ણન
left	લખાણને ખાનાની ડાબી તરફ ગોઠવવામાં આવે છે. સામાન્ય લખાણ માટે મોટાભાગે આ જ ગોઠવણી થાય છે.
right	લખાણને જમણી તરફ ગોઠવવામાં આવે છે.
center	લખાણ ખાનાની અંદર મધ્યમાં ગોઠવે છે.
justify	ખાનામાં રહેલા લખાણને એ રીતે ગોઠવવામાં આવે છે કે જેથી તે આખા ખાનામાં સમાઈ જાય.
char	ખાનાનું લખાણ કોઈ ચોક્કસ અક્ષરની પ્રથમ ઉપસ્થિતિ(instance)ની આસપાસ આડું ગોઠવાય છે. (ઉદાહરણ તરીકે, સંખ્યાઓ દશાંશ સ્થાનની પ્રથમ ઉપસ્થિતિને આધારે ગોઠવાય છે.)

કોષ્ટક 4.3 : align ગુણધર્મ માટે શક્ય કિંમતો

જો align ગુણધર્મની કિંમત char હોય તો આડી હરોળની અંદર આવેલા કોષ્ટકના દરેક ખાનાનું લખાણ કોઈ એક ચોક્કસ અક્ષરની પ્રથમ ઉપસ્થિતિની આસપાસ ગોઠવાશે. આવા નક્કી કરેલા ચોક્કસ અક્ષરને “ધરી અક્ષર” (axis character) તરીકે ઓળખવામાં આવે છે. આ ગુણધર્મ માટેનો આવો “ધરી અક્ષર” સામાન્ય સંજોગોમાં દશાંશ ચિહ્ન (.) હોય છે. લખાણની ગોઠવણી માટે char તરીકે દશાંશ ચિહ્ન (char=".")નો ઉપયોગ કરવાથી ખાનામાં રહેલી દશાંશ સંખ્યાઓ દર્શાવ્યા મુજબ દશાંશ સ્થાનના આધારે ગોઠવાશે.

	1	2	3	.	5	6	
		4	6	.	1	2	7
3	8	1	6	.	4	5	3

table ટેગ અને તેના ગુણધર્મો વિશે આટલી વિસ્તૃત ચર્ચા કર્યા પછી ચાલો હવે પછીના વિભાગમાં કોષ્ટકની આડી હરોળ માટેની <tr> ટેગ જોઈએ.

કોષ્ટકની આડી હરોળ માટેની ટેગ

કોષ્ટકમાં આડી હરોળ દર્શાવવા માટે <tr> ટેગનો ઉપયોગ કરવામાં આવે છે. <tr> ટેગની અંદર આવતી દરેક બાબત એ જ લીટી પર દર્શાવવી જોઈએ. તે કોષ્ટક 4.4માં દર્શાવ્યા મુજબ તેમાં ત્રણ ગુણધર્મો દર્શાવી શકાય.

ગુણધર્મ	વર્ણન
align	આડી હરોળનું લખાણ નક્કી કર્યા મુજબ ગોઠવાય છે.
bgcolor	પસંદ કરેલ આડી હરોળનો બેકગ્રાઉન્ડ રંગ નક્કી કરે છે.
valign	આડી હરોળના દરેક ખાનાના લખાણની ઊભી સમાંતર ગોઠવણી કરવાનું નક્કી કરે છે.

કોષ્ટક 4.4 : <tr> ટેગના ગુણધર્મો

તો ચાલો, હવે જોઈએ કે આ ગુણધર્મોનો HTML કોડમાં ઉપયોગ કેવી રીતે કરી શકાય.

Align ગુણધર્મ

align ગુણધર્મ આડી હરોળના બધા ખાનામાંના લખાણનું સ્થાન દર્શાવે છે. Align ગુણધર્મનું સામાન્ય સ્વરૂપ નીચે મુજબ છે :

align="alignment"

align ગુણધર્મની કિંમત તરીકે કોષ્ટક 4.3માં દર્શાવ્યા મુજબ left, right, center, justify અને char પૈકીની એક હોઈ શકે. કોષ્ટક રજૂ કરતા કોઈ પણ HTML કોડમાં આ ગુણધર્મોનો ઉપયોગ કરો.

Bgcolor ગુણધર્મ

અગાઉ કોષ્ટકના બેકગ્રાઉન્ડના રંગ પૂરવા આપણે bgcolor ગુણધર્મ જોયો, જેમાં આપણે રંગનું નામ લખીને અથવા રંગના સોળાંકી કોડ આપીને રંગ દર્શાવતા હતા. હવે જો કોષ્ટકની કોઈ એક આડી હારને ચોક્કસ રંગથી રંગવી હોય તો શું? ફરીથી આ માટેનો જવાબ છે ચોક્કસ રંગનું નામ અથવા નિયત સોળાંકી કોડ ધરાવતો bgcolor ગુણધર્મ. bgcolor ગુણધર્મ આડી હરોળ માટેનો રંગ ગોઠવે છે.

કોષ્ટકની એકાંતરે આપેલી આડી હરોળને જુદા જુદા રંગ વડે રંગવા <tr> ટેગ સાથે bgcolor ગુણધર્મ બહુ સામાન્યપણે ઉપયોગમાં લેવાય છે. આમ કરવાથી દરેક આડી હરોળને વાંચવામાં સરળતા રહે છે. કોડ-લિસ્ટિંગ 4.7માં આપેલ HTML કોડના ઉદાહરણનો પ્રયત્ન કરો.

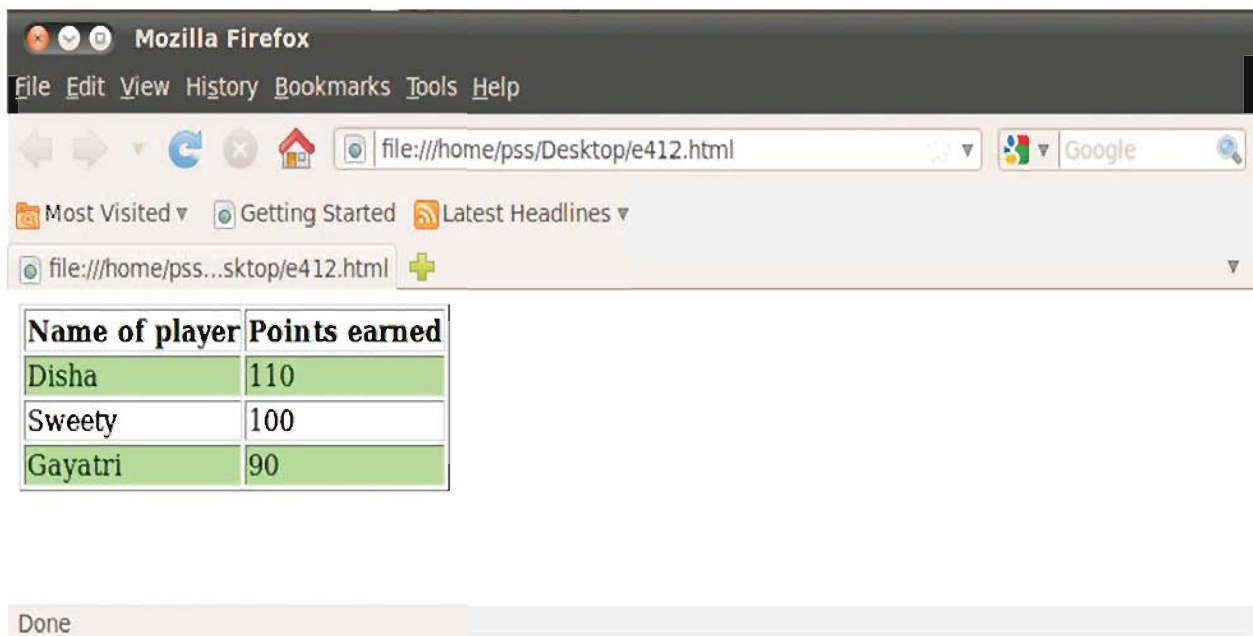
```

<html>
<body>
<table border="1">
<tr >
    <th>Name of player</th>
    <th>Points earned</th>
</tr>
<tr bgcolor="lightGreen">
    <td>Disha</td>
    <td>110</td>
</tr>
<tr >
    <td>Sweety</td>
    <td>100</td>
</tr>
<tr bgcolor="lightGreen">
    <td>Gayatri </td>
    <td>90</td>
</tr>
</table>
</body>
</html>

```

કોડ-વિસ્ટિંગ 4.7 : એક પછી એક આડી હરોળને આછા લીલા રંગે રંગવા માટેના HTML કોડ

કોડ-વિસ્ટિંગ 4.7નું પરિણામ આકૃતિ 4.11માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 4.11 : રંગબેરંગી બેકગ્રાઉન્ડ ધરાવતી આડી હરોળવાળું કોષ્ટક

Valign ગુણધર્મ

Valign ગુણધર્મ અહીં હરોળના દરેક ખાનાના લખાણની ઊભી સમાંતર ગોઠવણી કરવાનું નક્કી કરે છે. આવું નીચે મુજબની સામાન્ય વાક્યરચના(syntax)નો ઉપયોગ કરી શકાય.

valign="position"

કોષ્ટક 4.5, valign ગુણધર્મની શક્ય કિંમતો દર્શાવે છે.

કિંમત	વર્ણન
top	લખાણને ખાનાની ટોચના ભાગ તરફ ગોઠવે છે.
middle	લખાણને ખાનાની મધ્યમાં ગોઠવે છે.
bottom	લખાણને ખાનાની નીચે તરફ ગોઠવે છે.
baseline	લખાણને એ રીતે ગોઠવે છે કે, જેથી દરેક ખાનામાંના લખાણની પ્રથમ લીટી એ જ આડી સમાંતર લીટીથી શરૂ થાય.

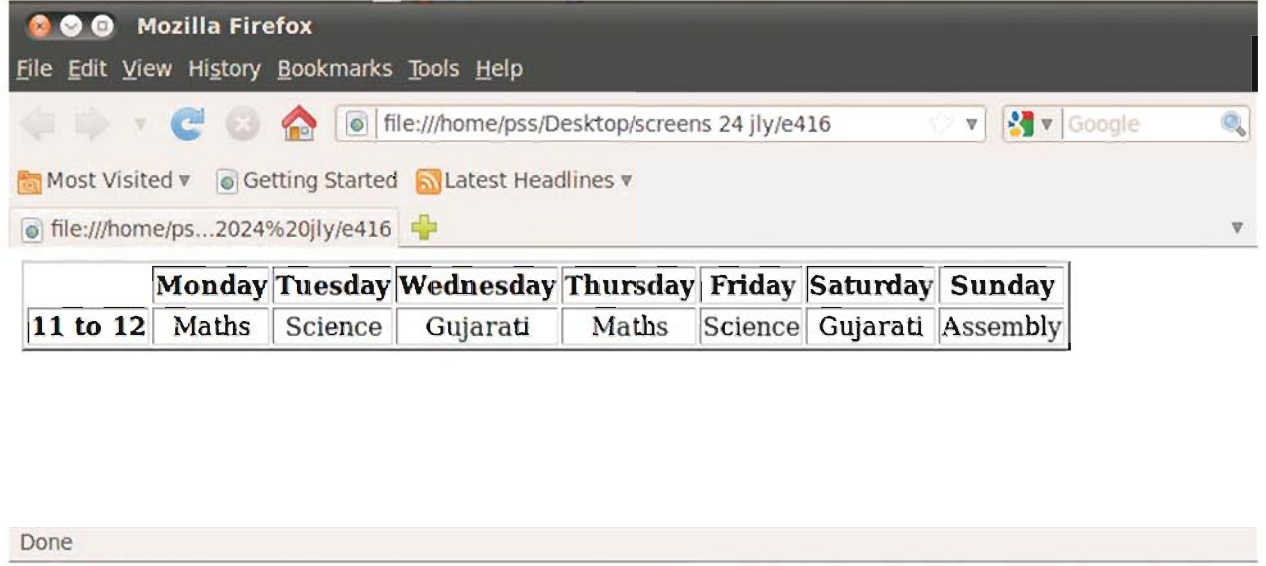
કોષ્ટક 4.5 : valign ગુણધર્મ માટેની શક્ય કિંમતો

<tr> ટેગનું ઉદાહરણ : <tr width="150". valign="bottom">. તમે પ્રયોગ કરતા હો તે HTML કોડમાં આ ટેગ લખીને બ્રાઉઝરમાં તેને ચકાસી જુઓ. કોડ-લિસ્ટિંગ 4.8માં આપેલ HTML કોડનો પ્રયત્ન કરી જુઓ.

```
<html>
<body>
<!-- ----->
    <table border="2">
        <tr>
            <th> </th>
            <th> Monday </th>
            <th> Tuesday </th>
            <th> Wednesday </th>
            <th> Thursday </th>
            <th> Friday </th>
            <th> Saturday </th>
            <th> Sunday </th>
        </tr>
<!-- ----->
        <tr align="middle">
            <th> 11 to 12 </th>
            <td> Maths </td>
            <td> Science </td>
            <td> Gujarati </td>
            <td> Maths </td>
            <td> Science </td>
            <td> Gujarati </td>
            <td> Assembly </td>
        </tr>
    </table>
<!-- ----->
</body>
</html>
```

કોડ-લિસ્ટિંગ 4.8 : સમયપત્રક તૈયાર કરવા માટેના HTML કોડ

કોડ-લિસ્ટિંગ 4.8માં દર્શાવેલ કોડ કોઈ એક વર્ગ માટે સાદા સમયપત્રકની બે આડી હરોળ બનાવે છે. જો તમે આ કોડને બ્રાઉઝરમાં જોશો તો તે આકૃતિ 4.12માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 4.12 : HTML કોડના ઉપયોગ દ્વારા બનતું સમયપત્રક

th અને tdના ઉપયોગ દ્વારા ખાનાઓની રજૂઆત

કોષ્ટકમાં દરેક ખાનું <td> અથવા <th> ટેગ દ્વારા રજૂ થાય છે. એ રીતે, આ બંને ટેગ કોઈ એક આડી હરોળને અનેક ખાનાઓમાં વિભાજિત કરવાની સવલત પૂરી પાડે છે. જો મથાળું નક્કી કરવાનું હોય તો <th> ટેગનો ઉપયોગ કરવામાં આવે છે, જ્યારે કોષ્ટકની વિગત રજૂ કરવી હોય તો <td> ટેગનો ઉપયોગ કરવામાં આવે છે.

સામાન્ય રીતે, <th> ટેગના લખાણને મોટેભાગે બોલ્ડ અક્ષરોમાં, ખાનાને આડા સમાંતર મધ્યમાં ગોઠવીને દર્શાવાય છે. જ્યારે <td> ટેગમાં સમાવિષ્ટ લખાણ સામાન્ય કદના અક્ષરોમાં ખાનાને સમાંતર ડાબી બાજુ ગોઠવીને દર્શાવાય છે. <th> અને <td> બંને ટેગમાં એક જ સરખા ગુણધર્મોનો ઉપયોગ થઈ શકે છે. આ ગુણધર્મની અસર માત્ર એક જ ખાના પર થશે. આ ગુણધર્મો દ્વારા ઊભી થનારી અસર આખા કોષ્ટક માટે કરેલી ગોઠવણી અથવા આની ઉપરના સ્તર જેમકે આડી હરોળ માટે કરેલી ગોઠવણીની અસરને પણ દબાવી દેશે. <th> અને <td> ગુણધર્મો દ્વારા થતી અસર એ જ આખરી ગણાશે.

સર્વ સામાન્ય ગુણધર્મો અને મૂળભૂત ઘટનાને લગતા ગુણધર્મો ઉપરાંત <th> અને <td> ટેગ સાથે કોષ્ટક 4.6માં દર્શાવ્યા મુજબના ગુણધર્મો પણ હોઈ શકે.

ગુણધર્મ	વર્ણન
abbr	ખાનાની માહિતીનું ટૂંકું સ્વરૂપ પૂરું પાડે છે.
align	ખાનાની માહિતીને ગોઠવે છે.
bgcolor	ખાનામાં બેકગ્રાઉન્ડ ઉમેરે છે.
char	ખાનાનાં લખાણને કોઈ ચોક્કસ અક્ષરની પ્રથમ ઉપસ્થિતિની આસપાસ ગોઠવે છે.
colspan	ખાનું કેટલાં ખાનાં સુધી વિસ્તરી શકે તે સંખ્યા દર્શાવે છે.
headers	ખાનાને સંબંધિત મથાળું દર્શાવે છે.
height	ખાનાની ઊંચાઈ નક્કી કરે છે.
nowrap	લખાણને ખાનાની અંદર જ આપમેળે નવી લાઈન ઉપર જતાં અટકાવે છે.
rowspan	ખાનું કેટલી આડી હરોળ સુધી વિસ્તરી શકે તે સંખ્યા દર્શાવે છે.
valign	ખાનામાં લખાણ ઊભી સમાંતર ગોઠવણી કરવાનું નક્કી કરે છે.
width	ખાનાની પહોળાઈ નક્કી કરે છે.

કોષ્ટક 4.6 : <td> અને <th>ના ગુણધર્મો

જેમ જેમ આપણે પ્રકરણમાં આગળ વધતાં જઈશું તેમ તેમ આપણે આ ગુણધર્મો વિશે વિગતવાર શીખીશું.

કોષ્ટકને નામ આપવું

કોષ્ટકને નામ આપવા માટે <caption> ટેગનો ઉપયોગ કરવામાં આવે છે. જ્યારે તમે કોષ્ટકના નામ સાથે કોઈ ચોક્કસ કોષ્ટક દર્શાવો ત્યારે આ ટેગની જરૂર પડે છે. કોષ્ટકનું નામ એ દર્શાવે છે કે, કોષ્ટક શેના માટે છે; પરિણામના હેતુ માટે, સમયપત્રક માટે અથવા ખોરાકની માત્રાનો આલેખ મોટાભાગના બ્રાઉઝર <caption> ટેગના લખાણને કોષ્ટકના ઉપરના ભાગે મધ્યમાં ગોઠવીને દર્શાવે છે.

ટેબલ બનાવતા કોઈ પણ માન્ય HTML કોડની પ્રથમ લીટીની ઉપર નીચે મુજબની લીટી ઉમેરવાથી કોષ્ટકની ઉપર "This is our timetable" એવું કોષ્ટકનું નામ દર્શાવશે.

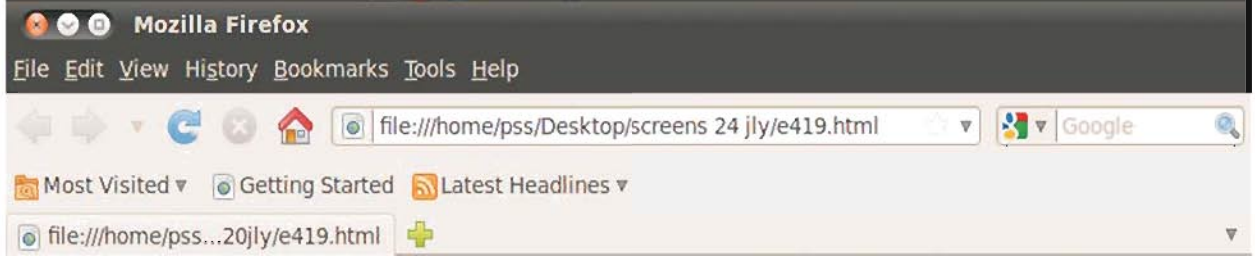
<caption> This is our timetable </caption>

કોડ-લિસ્ટિંગ 4.9 દર્શાવે છે કે, HTMLના ઉપયોગ દ્વારા કેવી રીતે સમયપત્રક બનાવી શકાય.

```
html>
<body>
  <table border="2">
    <!-- ----- -->
    <caption> <h1> <font color="Brown" >
      This is our time table
    </font> </h1> </caption>
    <!-- ----- -->
    <tr>
      <th> </th>
      <th> Monday </th>
      <th> Tuesday </th>
      <th> Wednesday </th>
      <th> Thursday </th>
      <th> Friday </th>
      <th> Saturday </th>
      <th> Sunday </th>
    </tr>
    <!-- ----- -->
    <tr align="middle">
      <th> 11 to 12</th>
      <td> Maths</td>
      <td> Science </td>
      <td> Gujarati</td>
      <td> Maths</td>
      <td> Science </td>
      <td> Gujarati</td>
      <td> Assembly </td>
    </tr>
  </table>
</body>
</html>
```

કોડ-લિસ્ટિંગ 4.9 : કોષ્ટકનું નામ ઉમેરવું

કોઈ પણ કોષ્ટકને નામ આપવાથી કોઈ પણ ઉપયોગકર્તા માટે કોષ્ટક પહેલી નજરમાં સમજાઈ જાય છે. આકૃતિ 4.13 કોડ-લિસ્ટિંગ 4.9નું પરિણામ રજૂ કરે છે.



This is our time table

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
11 to 12	Maths	Science	Gujarati	Maths	Science	Gujarati	Assembly

Done

આકૃતિ 4.13 : નામ સાથેનું કોષ્ટક

બરાબર ધ્યાનથી અવલોકન કરો, કે આકૃતિ 4.13માં દર્શાવાયેલ સમયપત્રક અપૂર્ણ છે. વિદ્યાર્થીઓ તેને કદાચ પૂર્ણ કરીને તેનું પરિણામનું અવલોકન કરી શકે.

કોષ્ટકની અંદર કોષ્ટક

ઘણીવાર આપણને કોઈ એક કોષ્ટકની અંદર બીજું કોષ્ટક ઉમેરવાની જરૂર પડતી હોય છે. કોડ-લિસ્ટિંગ 4.10માં દર્શાવેલ ઉદાહરણ કોષ્ટકની અંદર કોષ્ટક દર્શાવે છે.

```
<html>
<body>
<!-- ----- -->
<table border="4">
<caption> <h1>Conference Activities </h1></caption>
<tr>
  <th> </th>
  <th width="40%"> Morning </th>
  <th> Afternoon </th>
</tr>
<tr>
  <th> Day 1</th>
  <td> Inauguration </td>
  <td> Key-note Address</td>
</tr>
<!-- ----- -->
```

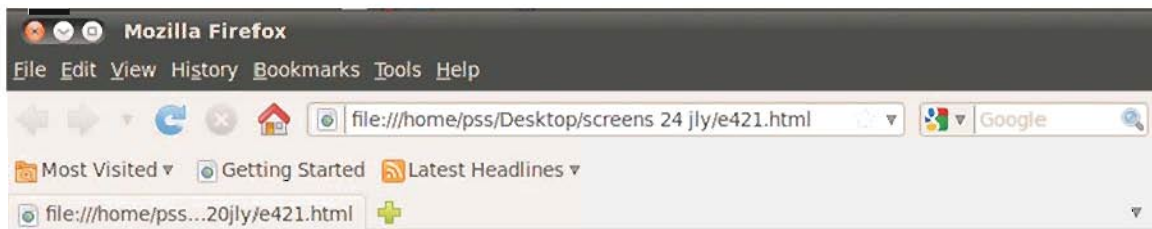
```

<tr>
  <th> Day 2 </th>
  <td > Paper Presentations</td>
  <td>
    <table border="1" bgcolor="pink" frame="box" align="right">
      <caption> <h3> Inner Table </h3></caption>
      <tr >
        <th> Teachers</th>
        <th> Students </th>
      </tr>
      <tr>
        <td> Meeting </td>
        <td> Quiz </td>
      </tr>
    </table>
  </td>
</tr>
</table >
<!-- ----->
</body>
</html>

```

કોડ-વિસ્ટિંગ 4.10 : કોષ્ટકની અંદર કોષ્ટક

આકૃતિ 4.14, કોડ-વિસ્ટિંગ 4.10નું પરિણામ દર્શાવે છે.



Conference Activities

	Morning	Afternoon
Day 1	Inauguration	Key-note Address
Day 2	Paper Presentations	Inner Table
		Teachers
Meeting	Quiz	

Done

આકૃતિ 4.14 : કોષ્ટકની અંદર કોષ્ટક

તમે કદાચ એ નોંધ્યું જ હશે કે અંદરના કોષ્ટકને બેકગ્રાઉન્ડ રંગ અને કિનારી છે. અંદરના તેમજ બહારના એમ બંને કોષ્ટકને તેમનાં નામ સાથે દર્શાવાયાં છે.

કોષ્ટકનું ઉદાહરણ

આકૃતિ 4.15 કોષ્ટકનું ઉદાહરણ દર્શાવે છે, તમે ક્યારેક તો જોયું જ હશે અને શક્યતઃ વાપર્યું પણ હશે. ભાવસહિત તમારી પસંદગીની વાનગીઓ ધરાવતું કોઈ પંજાબી રેસ્ટોરન્ટના મેનૂ વિશે છે.

Star Restaurant					
Starters		Roti and Bread		Curries	
Item	Price	Item	Price	Item	Price
Paneer Tikka Dry	100.00	Plain Roti	10.00	Paneer Bhurji	100.00
Panner kabab	100.00	Tandoori Roti	20.00	Mixed Veg	100.00
Drinks		Desserts		Water	
Item	Price	Item	Price	Item	Price
Tea	20.00	Gulab Jamun	30.00	Regular	---
Coffee	20.00	Rasgulla	35.00	Mineral	15
Koko juicy	25.00	Pudding	40.00	Sparkling	28
Mengo delight	25.00	Ice-cream	50.00		
The prices are in Indian rupees. Customers have to pay extra taxes. Please wait for 20 minutes after giving order.					

આકૃતિ 4.15 : રેસ્ટોરન્ટના મેનૂનું કાર્ડ

કોડ-લિસ્ટિંગ 4.11, આકૃતિ 4.15માં દર્શાવાયેલ રેસ્ટોરન્ટ મેનૂ કાર્ડ તૈયાર કરવા માટેના આંશિક HTML કોડ દર્શાવે છે.

```
<html>
  <head> <title> Star Restaurant Menu </title> </head>
  <body>
    <center>
<table width="75%" border="1" bordercolor="#000000" bordercolordark="#000000"
cellspacing="0">
  <tr align="center">
    <th align="center" colspan="6"> <h1> Star Restaurant </h1></th>
  </tr>
  <tr>
    <td colspan="2" align="center"> <h2> Starters </h2></td>
    <td colspan="2" align="center"> <h2> Roti and Bread </h2></td>
    <td colspan="2" align="center"> <h2> Curries </h2></td>
```

```

</tr>
<tr>
  <td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
  <td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
  <td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
  <td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
  <td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
  <td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
</tr>
<tr bgcolor="#CCCCCC">
  <td align="center"> <strong> Paneer Tikka Dry </strong> </td>
  <td align="center"> 100.00 </td>
  <td align="center"> <strong> Plain Roti </strong> </td>
  <td align="center"> 10.00 </td>
  <td align="center"> <strong> Paneer Bhurji </strong> </td>
  <td align="center"> 100.00 </td>
</tr>
<tr>
  <td align="center"> <strong> Paneer Kabab </strong> </td>
  <td align="center"> 100.00 </td>
  <td align="center"> <strong> Tandoori Roti </strong> </td>
  <td align="center"> 20.00 </td>
  <td align="center"> <strong> Mixed Veg </strong> </td>
  <td align="center"> 100.00 </td>
</tr>
<!-- --Remaining part of the table may be completed using similar code.... -->
<tr align="center">
  <td align="center" colspan="6"> The prices are in Indian rupees. <br>
    Customers have to pay extra taxes. <br>
    Please wait for 20 minutes after giving the order.
</td>
</tr>
</table>
</center>
</body>
</html>

```

કોડ-લિસ્ટિંગ 4.11 : HTML કોષ્ટકના ઉપયોગ દ્વારા રેસ્ટોરન્ટ મેનૂ કાર્ડ દર્શાવવું

અહીં એ નોંધ લો કે, કોડ-લિસ્ટિંગ 4.11માં આપેલ HTML કોડ આંશિક કોષ્ટક દર્શાવે છે. આવા જ વધુ HTML કોડ ઉમેરીને વિદ્યાર્થીઓ કોષ્ટકનો બાકીનો ભાગ પૂરો કરી શકે.

HTMLમાં ફ્રેમ

HTML દસ્તાવેજમાં ફ્રેમનો ઉપયોગ અનેક વેબપેઈજને ભેગાં કરીને એક જ વેબપેઈજ તરીકે દર્શાવવા થાય છે. ફ્રેમ, બ્રાઉઝરની વિન્ડોને ઘણા ભાગોમાં અથવા પેટા-વિન્ડોમાં વિભાજિત કરે છે જે દરેક એક અલગ વેબપેઈજ ધરાવે છે. બ્રાઉઝર વિન્ડોને અનેક ફ્રેમમાં વિભાજિત કરી શકાય છે, જેથી તમે જુદા જુદા HTML કોડને સ્વતંત્ર રીતે સંભાળી શકો અને તેને લઈ આવવાનું (loading) અને ફરી લઈ આવવાનું(reloading) કાર્ય સંભાળી શકો. બ્રાઉઝર વિન્ડોમાં ફ્રેમના સમૂહને “ફ્રેમસેટ” (frameset) તરીકે ઓળખવામાં આવે છે.

અહીં એ નોંધ લેવી જોઈએ કે, કેટલાક બ્રાઉઝર ફ્રેમસેટને સમર્થન આપતાં નથી. ફ્રેમસેટ દસ્તાવેજ તૈયાર કરવા, સૌપ્રથમ આપણે <frameset> એલિમેન્ટ બનાવવો પડે, જેને વિવિધ ફ્રેમના સંગ્રહક તરીકે ગણવામાં આવશે. ફ્રેમસેટ બ્રાઉઝર વિન્ડોનો વિભાગ વ્યાખ્યાયિત કરે છે. ફ્રેમસેટની અંદર દરેક ફ્રેમ <frame> અને </frame> ટેગ વડે દર્શાવવામાં આવે છે.

ફ્રેમસેટની અંદર, તમે <noframes> એલિમેન્ટ ઉમેરી શકો, જેથી જ્યારે ઉપયોગમાં લેવાતું બ્રાઉઝર ફ્રેમ સવલતને સમર્થન ન આપતું હોય ત્યારે તે ઉપયોગકર્તા માટે વૈકલ્પિક સંદેશ પૂરો પાડે છે. કોડ-લિસ્ટિંગ 4.12માં આપેલ ઉદાહરણ જુઓ.

```
<html>
<head>
<title>Creating example</title>
</head>
<frameset rows="20%,60%,20%">
<frame src="top.html" />
<frame src="main.html" />
<frame src="bottom.html" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

કોડ લિસ્ટિંગ 4.12 : ફ્રેમ બનાવવા માટેના HTML કોડ

કોડ-લિસ્ટિંગ 4.12માં દર્શાવેલ HTML કોડ લખો અને તેને frame.html તરીકે સંગ્રહ કરો. જરૂરી ફાઈલો બનાવો જેવી કે, top.html, main.html અને bottom.html. આ ફાઈલોમાં તમારું મનપસંદ લખાણ ઉમેરો. એકવાર તમે આ ફાઈલો બનાવી દો એ પછી આપણે તેને બ્રાઉઝરમાં ચકાસી શકીએ. આપણું પરિણામ આકૃતિ 4.16માં દર્શાવ્યા મુજબનું દેખાશે. અહીં એ ખાસ નોંધ લો કે, તમને જે પરિણામ જોવા મળશે તે તમે ફાઈલમાં શું લખાણ ઉમેર્યું છે તેના આધારે જુદું પડી શકે.

આકૃતિ 4.16માંથી એ અવલોકન કરવામાં આવ્યું કે, બ્રાઉઝર વિન્ડો ત્રણ ભાગમાં વિભાજિત કરવામાં આવી છે. વિન્ડોના ઉપરનો ભાગ વાદળી રંગે રંગવામાં આવે છે અને તે top.html નામની HTML ફાઈલનું લખાણ દર્શાવે છે. મધ્ય ભાગમાં રેસ્ટોરન્ટ મેનૂ માટે સાદું કોષ્ટક છાપે છે. મધ્ય ભાગ main.html નામની HTML ફાઈલ મારફત દર્શાવાય છે. આછા પીળા રંગે રંગાયેલો છેક નીચેનો ભાગ bottom.html નામની HTML ફાઈલ મારફતે દર્શાવાય છે.



આકૃતિ 4.16 : HTMLમાં ફ્રેમ

અહીં એ નોંધો કે, બ્રાઉઝર વિન્ડોને ઊભા ભાગમાં વહેંચવાનું પણ શક્ય છે. બ્રાઉઝર વિન્ડોને ઊભા ભાગમાં વહેંચવા માટે આપણે નીચે મુજબ ઊભી હરોળ (cols) સાથે frameset એલિમેન્ટનો ઉપયોગ કરી શકીએ.

<frameset cols="25%,*,25%">

અહીં, આપણે બ્રાઉઝર વિન્ડોને ત્રણ ઊભા ભાગમાં વહેંચવાનો પ્રયત્ન કરેલ છે. કોડ-લિસ્ટિંગ 4.12માં <frameset rows="20%,60%,20%"> કોડને બદલીને <frameset cols="25%,*,25%"> કરો અને ફરીથી તેનું પરિણામ નિહાળવાનો પ્રયત્ન કરો.

સારાંશ

આ પ્રકરણમાં, આપણે HTML માં યાદી અને કોષ્ટકો તૈયાર કરવાની વિવિધ રીતો જોઈ. આપણે જુદા જુદા ગુણધર્મો સહિત કમ્બાઇન યાદી ઉપરાંત અક્રમિક યાદી પણ જોઈ. યાદીની અંદર બીજી યાદી સમાવવાની વ્યવસ્થા પણ આ પ્રકરણમાં સામેલ કરવામાં આવી. સાદાં કોષ્ટકો ઉપરાંત કોષ્ટકની અંદર કોષ્ટક બનાવવાનું પણ આપણે શીખ્યા. આ પ્રકરણમાં, આખા કોષ્ટક પર અસર કરતા ગુણધર્મો ઉપરાંત અલગ કોષ્ટકની આડી હરોળ અને અલગ ખાનાંઓ માટેના ગુણધર્મો પણ જોયા.

સ્વાધ્યાય

1. યોગ્ય ઉદાહરણ આપીને સમજાવો કે HTML દસ્તાવેજમાં યાદી કેવી રીતે વ્યાખ્યાયિત કરવામાં આવે છે.
2. HTMLમાં વિવિધ પ્રકારની યાદી વર્ણવો.
3. HTMLમાં યાદીની અંદર યાદી ઉપર ટૂંક નોંધ લખો.
4. HTMLમાં કોષ્ટકના વ્યવસ્થાપન બાબતે ટૂંક નોંધ લખો.

5. નીચેના પ્રશ્નો માટે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ આપો :

- (1) HTMLમાં દ્વારા શું નક્કી થાય છે ?
(a) સાદી યાદી (b) ક્રમબદ્ધ યાદી
(c) અક્રમિક યાદી (d) સાદું કોષ્ટક
- (2) HTMLમાં દ્વારા નીચેનામાંથી કઈ બાબત નક્કી થાય છે ?
(a) સાદી યાદી (b) ક્રમબદ્ધ યાદી
(c) અક્રમિક યાદી (d) સાદું કોષ્ટક
- (3) યાદીમાંની વિગતોને નીચેનામાંથી કઈ ટેગ ઓળખી કાઢે છે ?
(a) અને (b) <items> અને </items>
(c) <object> અને </object> (d) <table> અને </table>
- (4) બુલેટ્સ વગર યાદી રજૂ કરવા માટે નીચેનામાંથી કઈ ટેગની જોડી ઉપયોગમાં લેવાય છે ?
(a) અને (b) અને
(c) <dt> અને </dt> (d) <pt> અને </pt>
- (5) કોષ્ટકની આડી હરોળ વ્યાખ્યાયિત કરવા માટે નીચેનામાંથી કઈ ટેગની જોડી ઉપયોગમાં લઈ શકાય ?
(a) <tr> અને </tr> (b) <td> અને </td>
(c) <col> અને </col> (d) <row> અને </row>
- (6) જ્યારે કોઈ ખાનું એક કરતાં વધુ આડી હરોળમાં વિસ્તરણ પામતું હોય ત્યારે નીચેનામાંથી કયો ગુણધર્મ ઉપયોગમાં લેવાય છે ?
(a) colspan (b) rowspan
(c) span (d) scope
- (7) બ્રાઉઝર વિન્ડોને અનેક ભાગોમાં વિભાજિત કરવા નીચેનામાંથી શાનો ઉપયોગ કરી શકાય ?
(a) Frameset (b) Elements
(c) Layouts (d) Design
- (8) જો બ્રાઉઝર frameને સમર્થન ન આપતું હોય તો, તેના બદલે વૈકલ્પિક લખાણ દર્શાવવા નીચેનામાંથી કયું ઉપયોગમાં લઈ શકાય ?
(a) Noframes (b) Yesframe
(c) Falseframe (d) Trueframe
- (9) HTMLની ક્રમબદ્ધ યાદીમાં નીચેનામાંથી કયું બદલી શકાય છે ?
(a) મુદ્દાઓનો ક્રમ (b) શરૂઆતનો ક્રમ
(c) ક્રમ માટે સંખ્યાચિહ્નનો પ્રકાર (d) આપેલા તમામ
- (10) HTML કોષ્ટકમાં નીચેનામાંથી કઈ વૈકલ્પિક બાબત (entity) છે ?
(a) કોષ્ટકનું નામ (Caption) (b) કોષ્ટકનું મથાળું (Heading)
(c) કોષ્ટકની ઊભી હરોળનું ગઠબંધન (d) આપેલા તમામ

પ્રાયોગિક સ્વાધ્યાય

1. તમારા મિત્રોનાં નામ અને શાળાની પરીક્ષામાં તેમણે મેળવેલ કુલ ગુણ (ટકાવારીમાં) દર્શાવતું એક કોષ્ટક તૈયાર કરતા HTML કોડ લખો.
2. તમે તમારા માતા-પિતાને આવક અને ઘરેલુ ખર્ચાની નોંધ રાખવા કાયદરીમાં હિસાબો લખતા જોયા હશે. ઘરેલુ કાર્ય માટે થતા ખર્ચા અને આવકો દર્શાવતું કોષ્ટક તૈયાર કરવા HTML કોડ લખો.
3. આ પ્રકરણના પ્રથમ વિભાગમાં દર્શાવાયા મુજબ ગુણપત્રક (marksheet) તૈયાર કરવા HTML કોડ લખો.
4. તમારા વર્ગનું સમયપત્રક તૈયાર કરવા માટેના HTML કોડ લખો. આ પ્રકરણમાં દર્શાવેલ સમયપત્રક નિહાળી યોજના ઘડો.
5. કોઈ એક મલ્ટિપ્લેક્સ સિનેમા માટે તેમાં ઉપલબ્ધ ચલચિત્રોની જાહેરાત તેમજ પ્રદર્શન થાય તે માટેનું કોષ્ટક તૈયાર કરવા માટેના HTML કોડ લખો. ધારી લો કે, તમારા શહેરના મલ્ટિપ્લેક્સ થિયેટરમાં ત્રણ સ્ક્રીન છે, અને સવારે 9 વાગે, બપોરે 1 વાગે, સાંજે 5 વાગે અને રાત્રે 9 વાગે એમ ચાર શો છે. ત્રીજા સ્ક્રીન પર મોડી રાત્રિનો (late night) શો ત્રણ જુદા જુદા પ્રાદેશિક ભાષાના ચલચિત્રો જેમાં એક ગુજરાતી, બીજું મરાઠી અને ત્રીજું તેલુગુ ઉપલબ્ધ બને છે. કોષ્ટકનો એક નમૂનારૂપ દેખાવ નીચે મુજબ છે. દરેક ખાનામાં તમારી પસંદગીના ચલચિત્રોના નામ ઉમેરો. અંદરના કોષ્ટકને "Regional Movies" એવું નામ પણ આપો.

Timings→	9:00 am	1:00 pm	5:00 pm	9:00 pm
Screen 1				
Screen 2				
Screen 3				Regional Movies

6. તમારે જે ચીજવસ્તુ ખરીદવી હોય તે વસ્તુઓની યાદી તૈયાર કરવા માટેના HTML કોડ લખો. ચીજવસ્તુઓને જુદા જુદા જૂથમાં વિભાજિત કરો જેમ કે, પુસ્તકો, શાકભાજી અને ખોરાકની ચીજો. નીચેનું નમૂનારૂપ ફેમવર્ક જુઓ, જે દર્શાવે છે કે યાદી કેવી હોવી જોઈએ.

List of items to be purchased

I. Books

- a. Textbook of science
- b. Practice book of maths
- c. Notebook

II. Vegetables

- a. Tomatoes
- b. Spinach
- c. Peas

III. Food items

- a. Chocolates
- b. Butter
- c. Bread



કેલ્સીનો પરિચય

કેલ્સી (calc) એ ઓપનઓફિસ સ્યૂટનું એક વિજાણુ વિસ્તારપત્રક (ઇલેક્ટ્રોનિક સ્પ્રેડશીટ) પેકેજ છે. સ્પ્રેડશીટ (Spreadsheet) એ પ્રકારનું સોફ્ટવેર પેકેજ છે જે દસ્તાવેજમાં ઝડપી અને સચોટ ગણતરીઓ કરવા માટે તેમજ દસ્તાવેજની માહિતીને આકર્ષક રીતે ગોઠવવા માટે (ફોર્મેટ કરવા) માટે વપરાય છે. (માહિતીને યોગ્ય તેમજ આકર્ષક રીતે ગોઠવવાની ક્રિયાને ફોર્મેટિંગ કહેવામાં આવે છે). આ સ્પ્રેડશીટ પેકેજનો ઉપયોગ નાણાંકીય અને હિસાબી દસ્તાવેજનાં વ્યવસ્થાપનમાં, માહિતીપત્રકો તૈયાર કરવામાં, મોકલેલા માલની કિંમત સાથેની યાદી (ઇન્વોઇસ) બનાવવામાં, વૈજ્ઞાનિક અને આંકડાકીય સંશોધનો પરથી માહિતી વિશ્લેષણ કરવામાં તેમજ માહિતી ઉપર વિવિધ પ્રકારની ગણતરીઓ કરવામાં થાય છે. કોઈ પણ સ્પ્રેડશીટ પ્રોગ્રામ ડેટાનો સંગ્રહ કરવા, તેના ઉપર ગણતરીઓ કરવા તેમજ માહિતીને રેખાત્મક રીતે (ગ્રાફિકલ) રજૂઆત કરવા સમક્ષ હોય છે. સ્પ્રેડશીટ પેકેજ ડેટાને આડી હરોળ (row) અને ઊભી હરોળ(column)નાં સ્વરૂપમાં દાખલ કરવાની પરવાનગી આપે છે. તમે યાદ કરો : ગણિતનો અભ્યાસ કરવા માટે કદાચ તમે તમારા બાળપણમાં ખાસ પ્રકારની નોટબુક કે જેમાં આડી હરોળ અને ઊભી હરોળ વડે ચિહ્નિત સેલ હોય તેનો ઉપયોગ કર્યો હશે. સ્પ્રેડશીટ એ ડેટાનું વિશ્લેષણ તેમજ તેના ઉપર ગણતરીઓ કરવા માટેની કમ્પ્યુટર સ્ક્રીન ઉપરની આડી હરોળ અને ઊભી હરોળ વડે બનેલી એક લાંબી શીટ (વિસ્તાર) છે. બીજી રીતે કહીએ તો સ્પ્રેડશીટ એ એક ગ્રીડ (જાળી જેવું બંધારણ) છે કે જે ક્રિયા-પ્રતિક્રિયાની રીતે (interactively) ડેટાનું રો અને કોલમની રીતે સંચાલન તેમજ આયોજન કરે છે. આ કામગીરીની સરળતા માટે સ્પ્રેડશીટ પેકેજ ડેટા અને ફોર્મેટિંગની સગવડ ઉપરાંત સૂત્રો દાખલ કરવાની પરવાનગી પણ આપે છે. કોઈ નિર્ણય લેવા માટે ઉપયોગકર્તા સ્પ્રેડશીટનાં પાનાંમાં ક્રિયા-પ્રતિક્રિયાની રીતે ડેટા દાખલ કરી શકે, ડેટાને ફોર્મેટ કરી શકે તેમજ ડેટા ઉપર ગણતરી / વિશ્લેષણ કરી શકે છે. ઉપયોગકર્તાની માહિતી અને સૂત્રો ઉપરાંત સર્વસામાન્ય ગાણિતિક, નાણાંકીય, આંકડાકીય અને તાર્કિક કાર્યો ખૂબ જ અત્યાધુનિક રીતે કરવા માટેનાં સૂત્રો / વિધેયો સ્પ્રેડશીટ પેકેજમાં જ સમાવિષ્ટ કરેલાં હોય છે. સ્પ્રેડશીટ પેકેજની આ પ્રકારની અનેક ક્ષમતાઓને કારણે સંગઠિત (structured) ડેટા તૈયાર કરવા તેમજ તેના ઉપર પ્રક્રિયા કરવા માટે તે એક વિશ્વવ્યાપી પ્રોગ્રામ તરીકે વપરાય છે. Libre Office Calc [www.libreoffice.org] પ્રોગ્રામ એ calc જેવો લાગે છે, અને તે નિઃશુલ્ક અને ઓપનસોર્સ સ્પ્રેડશીટ પેકેજ પણ છે.

સ્પ્રેડશીટની પેલી પાર (Beyond a spreadsheet)

જેમ જેમ સ્પ્રેડશીટનું કદ મોટું બનતું જાય છે તેમ તેમ તેના સંચાલનનું કાર્ય વધારે કઠિન બને છે. સ્પ્રેડશીટનાં વધતા કદનાં નિયંત્રણ માટે વર્કબુક(workbook)નો એક ખ્યાલ અમલમાં મૂકવામાં આવ્યો. વર્કબુકનો મુખ્ય હેતુ સંખ્યાબંધ સ્પ્રેડશીટનું સારી રીતે સંચાલન કરવાનો છે. વર્કબુક ઉપરાંત હાલમાં અનેક ઓનલાઈન સ્પ્રેડશીટ પણ લોકપ્રિય બની રહી છે.

કેટલાંક લાક્ષણિક વિનિયોગોમાં સ્પ્રેડશીટ પેકેજનો ઉપયોગ (Typical Applications of Spreadsheet Packages)

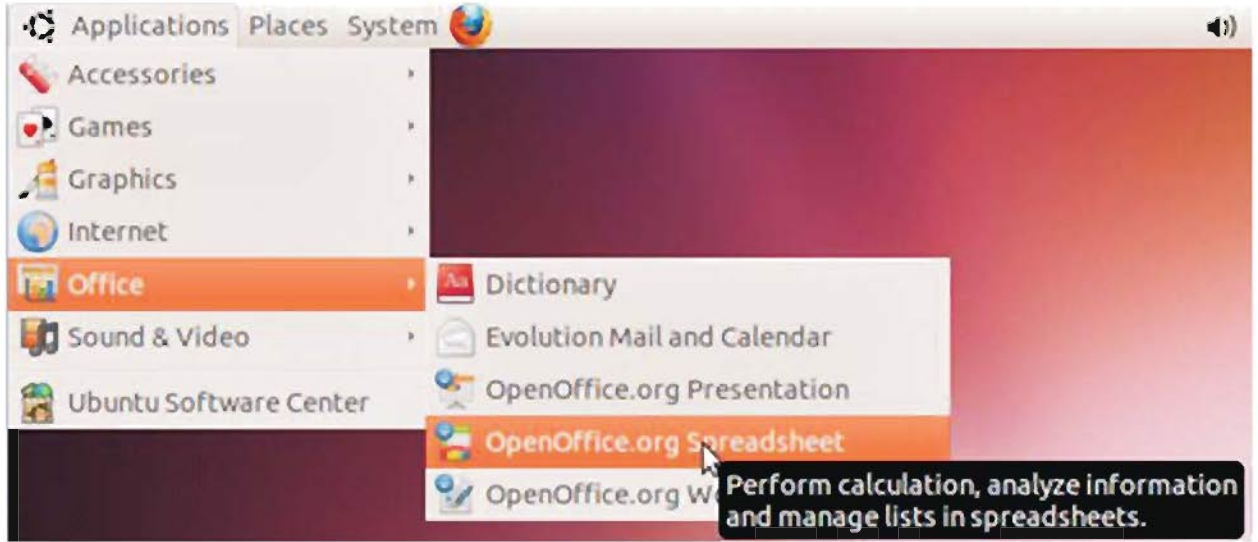
માહિતી વિશ્લેષણ તેમજ એકાઉન્ટિંગનાં કાર્યો માટે સ્પ્રેડશીટ પેકેજનો વ્યાપકપણે ઉપયોગ થાય છે. સામાન્ય રીતે સ્પ્રેડશીટ પેકેજ દ્વારા કરવામાં આવતાં કેટલાંક વિશિષ્ટ કાર્યોની યાદી કોષ્ટક 5.1માં આપવામાં આવેલી છે.

કાર્ય	વર્ણન
સરવૈયું (બેલેન્સ શીટ)	નાણાંકીય સ્થિતિનાં પત્રકો અને લાક્ષણિક એકાઉન્ટ ફોર્મેટમાં સારાંશનાં પત્રકો
પરિણામ વિશ્લેષણ અને મેરિટની યાદી બનાવવી	પરીક્ષાઓ જેવી કેટલીક પ્રવૃત્તિઓનાં પરિણામની ગણતરી, સોર્ટિંગ (ક્રમબદ્ધ ગોઠવણી) તેમજ ફિલ્ટરીંગ (કોઈ ચોક્કસ ધોરણો આધારિત પરિણામ)
આંકડાકીય ડેટા વિશ્લેષણ	સોર્ટિંગ (ક્રમમાં ગોઠવણી), ડેટાની વ્યવસ્થિત ગોઠવણી અને આંકડાકીય પદ્ધતિઓનો અમલ જેમ કે મધ્યક (mean), મધ્યસ્થ (median) અને સંભાવના (probability) વગેરેની ગણતરી કરવી
નાણાંકીય કાર્યો	લોનના હપ્તાની ગણતરી, વ્યાજની ગણતરી વગેરે જેવાં નાણાંકીય કાર્યો
અંગત કાર્યો	વજનનું વ્યક્તિગત રીતે નિયંત્રણ (દિબરેખ), વિવિધ પ્રવૃત્તિઓ, વસ્તુઓ અને ઘટનાની યાદીનું સંચાલન જેમ કે કોઈ મિજબાની અથવા આગામી ઘટના માટે મહેમાનોની યાદી બનાવવી

કોષ્ટક 5.1 : સ્પ્રેડશીટના કેટલાક પ્રચલિત વિનિયોગો

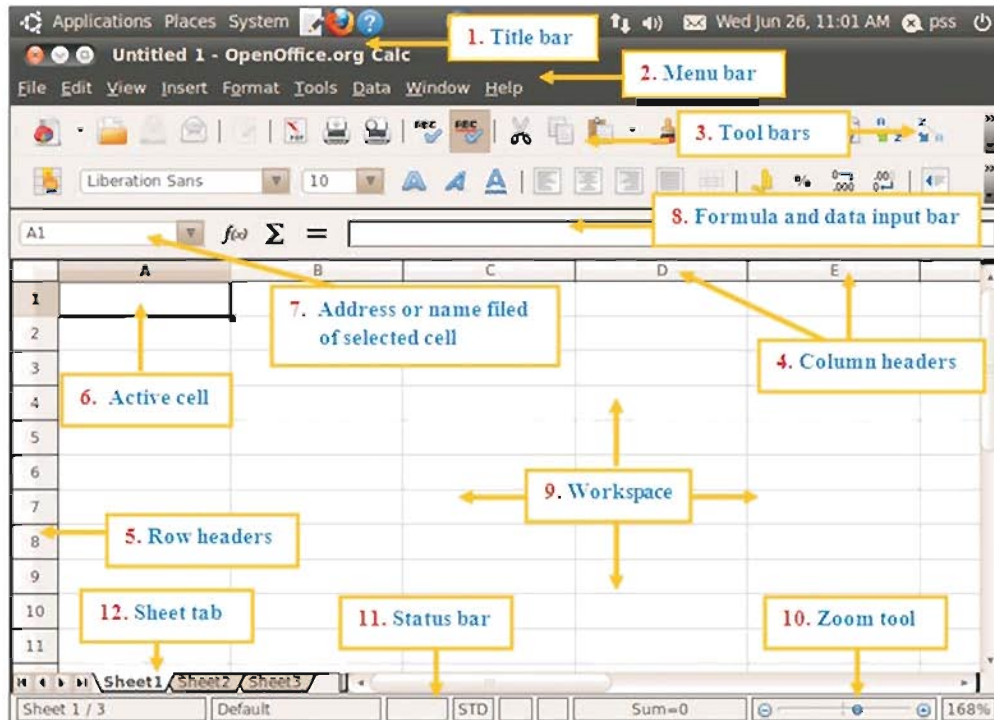
કેલ્સી સાથે પ્રારંભ (Getting started with Calc)

સૌપ્રથમ તમારે એ જાણવું જોઈએ કે કેલ્સીને કઈ રીતે ખોલી શકાય. આ માટે તમે આકૃતિ 5.1માં દર્શાવ્યા પ્રમાણેના આદેશોની શ્રેણીને અનુસરો : **Applications → Office → OpenOffice.org Spreadsheet**



આકૃતિ 5.1 : કેલ્સી સાથે શરૂઆત

આ આદેશ એક વિનિયોગ દસ્તાવેજ ખોલશે જે સ્પ્રેડશીટ કહેવાય છે. સ્પ્રેડશીટ અનેક શીટની બનેલી હોય છે; જે વર્કશીટ (worksheet) પણ કહેવાય છે. પૂર્વનિર્ધારિત રીતે નવી સ્પ્રેડશીટમાં ત્રણ વર્કશીટ હોય છે પણ આપણી જરૂરિયાત પ્રમાણે તેમાં સંખ્યાબંધ નવી વર્કશીટ ઉમેરી પણ શકાય તેમજ દૂર પણ કરી શકાય છે. કોઈ પણ કેલ્સી સ્પ્રેડશીટ દસ્તાવેજમાં મહત્તમ 256 વર્કશીટ હોઈ શકે.



આકૃતિ 5.2 : એક ખાલી વર્કશીટ

જ્યારે તમે કેલ્સીમાં એક નવો દસ્તાવેજ ખોલો છો ત્યારે આકૃતિ 5.2માં દર્શાવ્યા પ્રમાણે એક ખાલી સ્પ્રેડશીટ તમે સ્ક્રીન ઉપર જોશો. તેમાં નંબર ટેગ સાથે અલગ-અલગ ઘટક દર્શાવેલ છે. ચાલો, હવે આપણે દરેક ઘટકની વિગતવાર ચર્ચા કરીએ.

ટાઈટલ બાર (Title bar)

ટાઈટલ બાર સૌથી ઉપર સ્થિત થયેલ હોય છે. તે હાલમાં ખુલ્લી સ્પ્રેડશીટનું નામ (શીર્ષક) બતાવે છે. જો તમે હમણાં નવી જ ફાઈલ ખોલી છે અને સ્પ્રેડશીટને નામ આપેલું નથી તો તે એક નામ વગરની સ્પ્રેડશીટ છે. તે Untitled Xથી દર્શાવવામાં આવે છે, જ્યાં X એ એક સંખ્યા છે. જ્યારે કોઈ સ્પ્રેડશીટ તમે પ્રથમ વખત સંગ્રહ કરો છો ત્યારે તમારી પસંદગીનું નામ તમને પૂછવામાં આવે છે. તે તમને હાલમાં કઈ વર્કબુકમાં કામ કરી રહ્યા છો તે જણાવશે. આકૃતિ 5.3માં ટાઈટલ બાર દર્શાવેલ છે.



આકૃતિ 5.3 : ટાઈટલ બાર

મેનૂ બાર (Menu bar)

મેનૂ બારમાં વિવિધ કાર્યો કરવા માટેના મેનૂ સાથે આદેશો હોય છે. દરેક મેનૂ આઈટમ એક અલગ સબમેનૂ (submenu) રજૂ કરે છે. આ સબમેનૂને પુલડાઉન મેનૂ (pull down menu) પણ કહેવામાં આવે છે. મેનૂ જોવા માટે માઉસ કર્સરને ફક્ત તેના ઉપર મૂકો અને એક વાર ક્લિક કરો આથી સબમેનૂ આપણને સ્ક્રીન ઉપર દેખાય છે. આકૃતિ 5.4માં એક લાક્ષણિક મેનૂ બાર દર્શાવેલ છે.



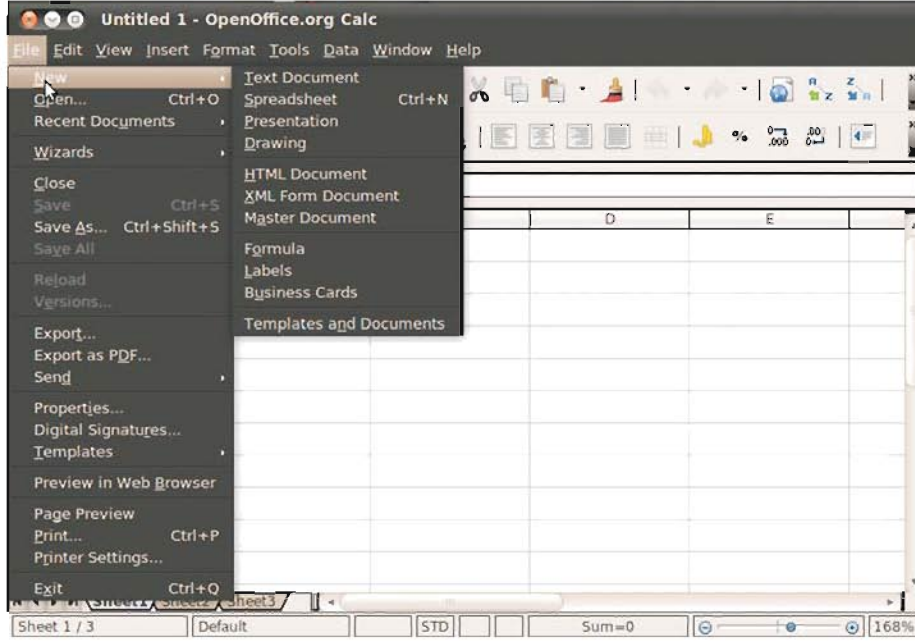
આકૃતિ 5.4 : મેનૂ બાર

મુખ્ય મેનૂના દરેક આદેશોની તેનાં ટૂંકા વર્ણન સાથેની યાદી કોષ્ટક 5.2માં આપેલી છે.

મેનૂ	વર્ણન
File	દસ્તાવેજમાં કામ કરવા માટેના આદેશો જેવા કે નવા દસ્તાવેજ બનાવવા, ફાઈલનો સંગ્રહ કરવો, પ્રિન્ટિંગ કરવું, પ્રિન્ટ કરતાં પહેલાં સ્ક્રીન ઉપર તેનો દેખાવ જોવો (પ્રિન્ટ પ્રિવ્યુ) વગેરે.
Edit	દસ્તાવેજમાં સુધારા-વધારા કરવા માટેના આદેશો જેવા કે copy, paste, find & replace વગેરે.
View	વપરાશકર્તાના સેતુના ઘટકો ઉમેરવા અથવા દૂર કરવા, પાનું પૂરું થાય તેની નિશાની જોવી (પેઈજ બ્રેક પ્રિવ્યુ) વગેરે.
Insert	આડી હાર, ઊભી હાર, વર્કશીટ, ઘટકો અને ઓબ્જેક્ટ ઉમેરવા વગેરે.
Format	સેલ ફોર્મેટ કરવા, ઘટકોનો સમૂહ બનાવવો, સોર્ટિંગ, શરતી ફોર્મેટિંગ વગેરે.
Tools	વધારાનાં ટૂલ્સ જેવાં કે સ્પેલિંગ ચેક કરવા માટે, દસ્તાવેજનું રક્ષણ કરવા માટે, સૂત્રો તેમજ ભૂલો સુધારવા માટે વગેરે.
Data	ડેટા પર પ્રક્રિયા કરવી, ડેટા સોર્ટ કરવો, શરત પ્રમાણે માત્ર જરૂરી ડેટા રજૂ કરવો (ફિલ્ટર) વગેરે.
Window	નવી વિન્ડો ખોલવી, સેલને સ્થાયી સ્વરૂપ આપવું તેમજ OpenOffice.orgના ખુલ્લા દસ્તાવેજોની યાદી વગેરે.
Help	વિધેય તેમજ વિનિયોગ બાબતની માહિતી અને સોફ્ટવેરની આવૃત્તિઓ વગેરે બાબતની મદદ

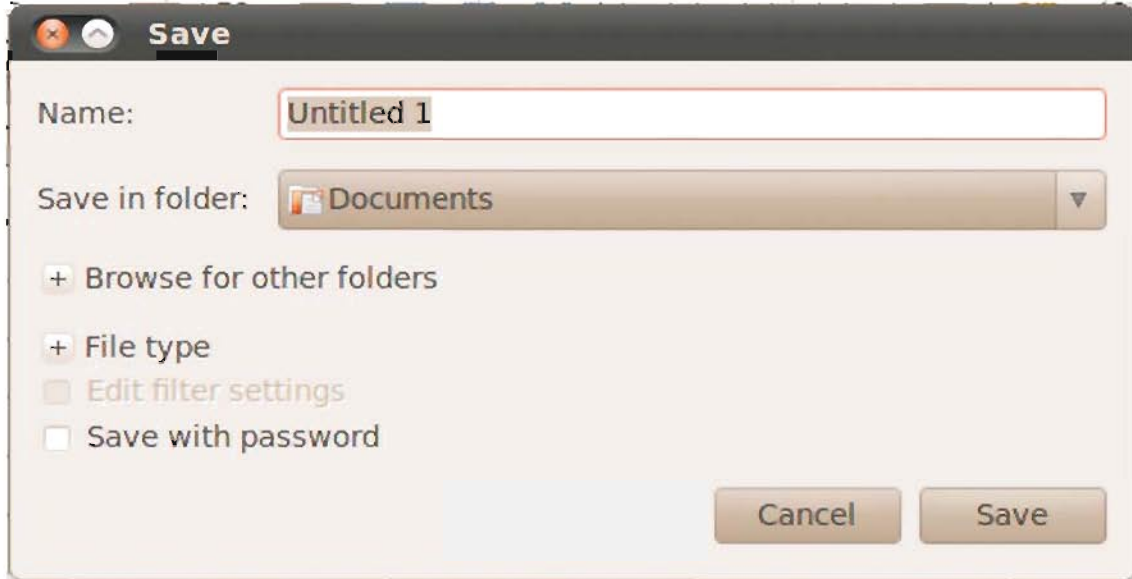
કોષ્ટક 5.2 : કેલ્સીના વિવિધ મેનૂ

જો તમે કોઈ એક મેનૂ પસંદ કરશો તો તેને અનુરૂપ સબમેનૂ સ્ક્રીન ઉપર પ્રદર્શિત થશે. સબમેનૂના વિકલ્પો સાથે એક ત્રિકોણ (►) આકાર કદાચ તમારા ધ્યાન ઉપર આવ્યો હશે. જ્યારે આ પ્રકારની ત્રિકોણ (►) સંજ્ઞા સ્ક્રીન ઉપર પ્રદર્શિત થાય ત્યારે તે વિકલ્પમાં વધારાના વિકલ્પો ઉપલબ્ધ હોય છે; જેમાંથી એક યોગ્ય વિકલ્પની પસંદગી તમે કરી શકો છો. આકૃતિ 5.5માં આ પ્રકારની સ્થિતિ દર્શાવેલ છે. આકૃતિ 5.5માં File મેનૂની અંદર 'New' વિકલ્પની પસંદગી સાથે ત્રિકોણ (►) આકાર વધારે વિકલ્પોને સૂચવે છે. આ વિકલ્પો બીજી કાળી ઊભી હરોળનાં સ્વરૂપમાં પોપ ડાઉન મેનૂ તરીકે રજૂ કરવામાં આવે છે.



આકૃતિ 5.5 : મેનૂના ઉપલબ્ધ વિકલ્પો

જો ઉપયોગકર્તાની સૂચનાઓનું પાલન કરવા માટે વધારાની માહિતીની જરૂરિયાત હોય તો (...) સંજ્ઞા દર્શાવતું એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થશે. ધારો કે તમે **Save as** વિકલ્પ પસંદ કરો છો તો આકૃતિ 5.6માં દર્શાવ્યા પ્રમાણે કેલ્સી તમને ફાઈલનું નામ અને કઈ જગ્યાએ તે ફાઈલનો સંગ્રહ કરવો છે તેની માહિતી તમને પૂછશે.



આકૃતિ 5.6 : Save ડાયલોગ બોક્સ

એ જ રીતે તમે **"Print"** વિકલ્પ નિહાળશો તો **"Print"** વિકલ્પનાં અંતમાં બિંદુઓની સંજ્ઞા (...) પણ તમે જોઈ શકશો; જેનો અર્થ એ છે કે અનેક પ્રિન્ટ વિકલ્પો દર્શાવતું એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થશે. આપણે આ બધા મેનૂ વિશે ઊંડાણમાં ચર્ચા હવે પછી કરીશું.

ટૂલબાર (Toolbar)

સૌથી સામાન્ય આદેશોને ટૂલબાર સ્ક્રીન ઉપર જુદા જુદા બટન સ્વરૂપમાં રજૂ કરે છે. માઉસ પોઈન્ટરને ફક્ત તે બટન ઉપર ક્લિક કરીને તમે તે યુટિલિટીને પસંદ કરી શકો છો. કદાચ જો કોઈ બટન શું કાર્ય કરે છે તે તમે ન જાણતા હો તો તમારે ફક્ત તે બટન ઉપર માઉસ પોઈન્ટરને આમતેમ ફેરવવું પડશે. આથી તરત જ તે બટનને અનુરૂપ કાર્યનું નામ સ્ક્રીન ઉપર પ્રદર્શિત થશે. **View → Toolbars** પસંદ કરીને તમે તમારી જરૂરિયાત પ્રમાણે ટૂલબારને પ્રદર્શિત કરી પણ શકો અને પ્રદર્શિત થતા બંધ પણ કરી શકો. પૂર્વનિર્ધારિત રીતે (સિસ્ટમ દ્વારા સેટ કરેલો) ટૂલબાર નીચે પ્રમાણે છે :

- **સ્ટાન્ડર્ડ ટૂલબાર (Standard toolbar) :** ટૂલબારનું નામ સૂચવે છે તે પ્રમાણે, આ ટૂલબાર ઉપર File અને Edit મેનૂમાંથી વારંવાર વપરાતા આદેશો રાખેલા હોય છે. તાજેતરમાં ઉપયોગમાં લેવાયેલા આઈકોન આ બાર ઉપર જોવા મળે છે. સામાન્ય રીતે નવો કેલ્સી દસ્તાવેજ બનાવવા, હયાત દસ્તાવેજ ખોલવા, દસ્તાવેજમાં લખાણની જોડણી ચકાસવા, માહિતી કટ અને પેસ્ટ કરવા અને દસ્તાવેજ પ્રિન્ટ કરવા વગેરે માટેના આઈકોન તમે આ સ્ટાન્ડર્ડ ટૂલબાર ઉપર જોશો, આકૃતિ 5.7માં એક સ્ટાન્ડર્ડ ટૂલબાર દર્શાવેલ છે.



આકૃતિ 5.7 : સ્ટાન્ડર્ડ ટૂલબાર

- **ફોર્મેટિંગ ટૂલબાર (Formatting toolbar) :** સેલની માહિતીને ફોર્મેટ કરવા માટેનાં સૌથી વધારે વપરાતા આદેશો ફોર્મેટિંગ ટૂલબાર ઉપર હોય છે. આ ટૂલબાર ઝડપથી ફોર્મેટિંગ કરવાની સગવડ પૂરી પાડે છે. જેમ જેમ તમે વિનિયોગનો ઉપયોગ કરતા જશો તેમ તેમ આ બાર ઉપરનાં આઈકોન બદલાતાં જશે. જો કોઈ આઈકોન આ બાર ઉપર ઉપલબ્ધ ન હોય તો ફોર્મેટિંગ ટૂલનાં બિલકુલ અંતમાં જઈને તમે આઈકોન ઉમેરવાનો વિકલ્પ પસંદ કરી શકો. આકૃતિ 5.8માં એક લાક્ષણિક ટૂલબાર દર્શાવેલ છે.



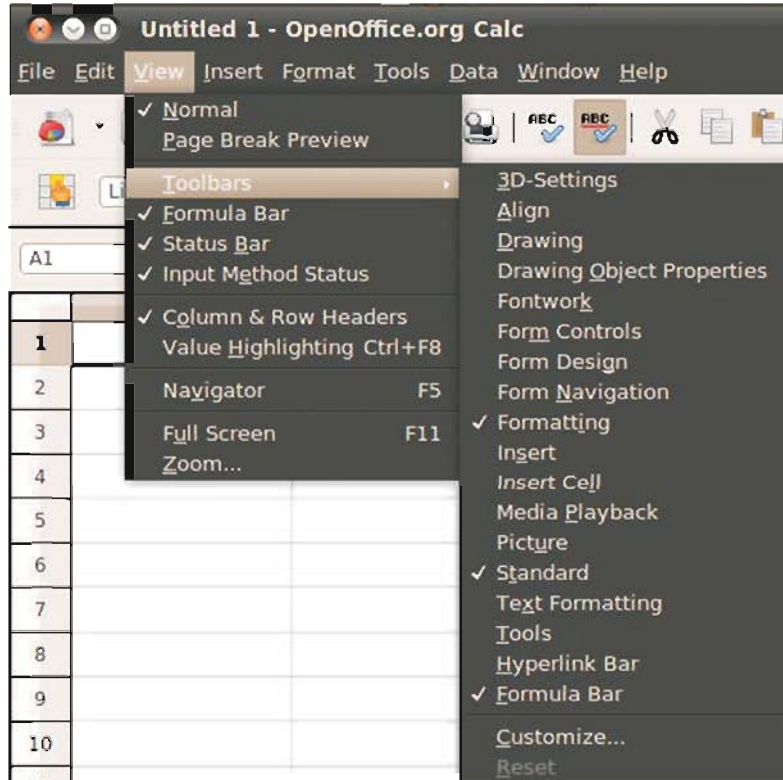
આકૃતિ 5.8 : ફોર્મેટિંગ ટૂલબાર

- **ફોર્મ્યુલા ટૂલબાર (Formula toolbar) :** સેલમાં કોઈ સૂત્ર લખવા અથવા તેમાંની માહિતી સુધારવાની સગવડ ફોર્મ્યુલા ટૂલબાર ઉપર હોય છે. આકૃતિ 5.9માં એક લાક્ષણિક ફોર્મ્યુલા ટૂલબાર દર્શાવેલ છે.



આકૃતિ 5.9 : ફોર્મ્યુલા ટૂલબાર

ટૂલબાર ઉપર રહેલાં બટન બદલી શકાય છે. ઉપયોગકર્તા પોતાની ઇચ્છા પ્રમાણે ટૂલબાર ઉપર બટન ઉમેરી શકે છે, બટન દૂર કરી શકે છે તેમજ બટનની શ્રેણી (ગોઠવણી) બદલી પણ શકે છે. આકૃતિ 5.10માં દર્શાવ્યા પ્રમાણે તમે ટૂલબાર ઉમેરી શકો તેમજ દૂર પણ કરી શકો છો.

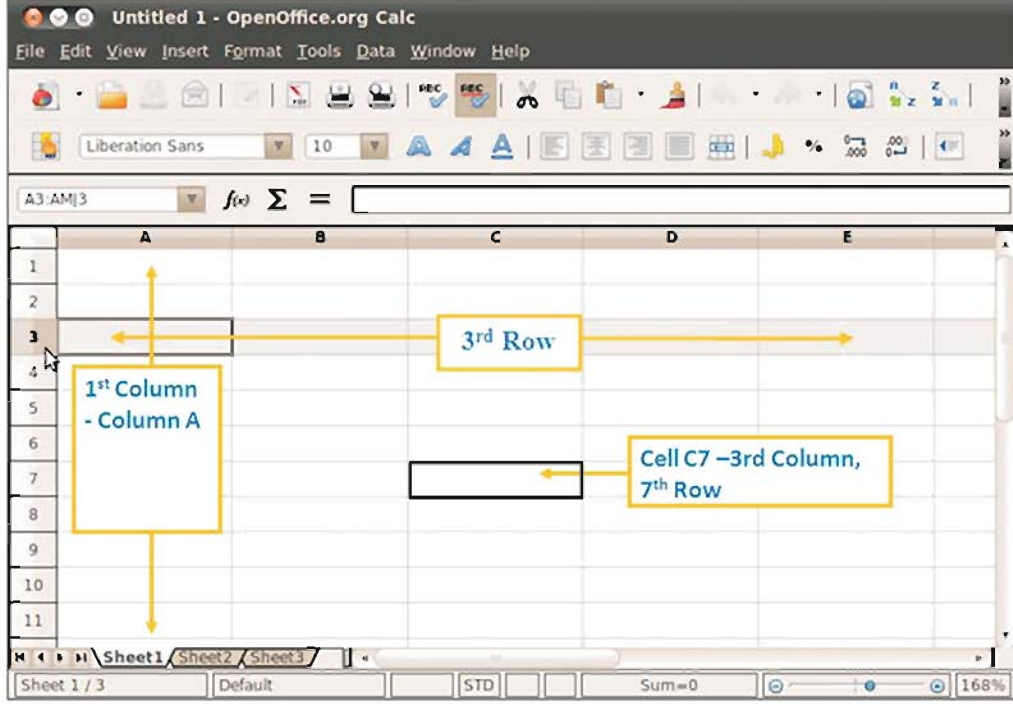


આકૃતિ 5.10 : ટૂલબાર ઉમેરવા / દૂર કરવા

આડી હરોળ, ઊભી હરોળ અને સેલ (Rows, columns and cells)

દરેક વર્કશીટને ઊભી હરોળ (કોલમ) અને આડી હરોળ(રો)માં વિભાજિત કરીને સેલ બનાવવામાં આવે છે. દરેક રો અને કોલમને ક્રમ આપવામાં આવે છે; કોલમને મૂળાક્ષરો (અને તેનાં સંયોજનો) અને રોને સંખ્યાથી ઓળખવામાં આવે છે. વર્કશીટની ત્રીજી રો આકૃતિ 5.11માં હાઈલાઈટ થયેલી છે.

કેલ્સીની દરેક કોલમ સેલની ઊભી શ્રેણી છે. એડશીટના યોગ્ય અક્ષર ઉપર ક્લિક કરીને આખી કોલમ પસંદ કરી શકાય છે. આકૃતિ 5.11માં વર્કશીટની પહેલી કોલમ દર્શાવેલ છે.



આકૃતિ 5.11 : કેલ્સીની રો

સેલ એ રો (આડી હરોળ) અને કોલમ(ઊભી હરોળ)નું છેદનબિંદુ છે. (રિખાઓ અરસપરસ છેદવાથી બનતો વિસ્તાર છે.) સેલને કોલમનાં મૂળાક્ષર (મૂળાક્ષરો) અને રો નંબરનાં સંયોજનથી ઓળખવામાં આવે છે, દા.ત. સેલ C7. આકૃતિ 5.11માં સેલ C7 દર્શાવેલ છે, જે સેલ હાલમાં પસંદ કરેલ હોય તેને કાર્યરત (એક્ટિવ) સેલ કહેવામાં આવે છે. અહીં આકૃતિમાં સેલ C7 ને કાર્યરત સેલ બતાવ્યો છે. (અહીં એ નોંધ કરશો કે વાસ્તવિક સ્ક્રીનના કાર્યરત સેલની જમણી બાજુની નીચે વત્તાનું ચિહ્ન હોય છે.) અહીં જે ચોરસ બોક્સ દેખાય છે તે કાર્યરત સેલનું સૂચક છે. કાર્યરત સેલના ખૂણામાં ભરેલો એક નાનો લંબચોરસ હાઈલાઈટ થયેલ છે જે ઓટોફિલ હેન્ડલ (autofill handle) તરીકે ઓળખાય છે, જેની ચર્ચા આપણે હવે પછી આગામી પ્રકરણમાં કરીશું.

આપણી પાસે સરખામણીમાં અંગ્રેજી મૂળાક્ષરો (26) થોડા અને વધારે સંખ્યામાં રો હોવાથી આપણને મૂળાક્ષરોનાં સંયોજનનો ઉપયોગ કરવાની જરૂર પડે છે જેમ કે AA, AB, ..., AZ, BA, ..., BZ, CA, વગેરે. કેલ્સીમાં 1024 કોલમ અને 1,048,576 રો હોય છે.

સેલ એ એડશીટનો એક પાયારૂપ મૂળભૂત ભાગ છે. સેલમાં ડેટા અને સૂત્રો ઉમેરવામાં આવે છે. એટલે કે, સેલ અલગ અલગ ઘટકો જેવા કે શાબ્દિક માહિતી, સંખ્યા અને સૂત્રો ધરાવે છે. તમે કોઈ એક સેલ ઉપર ક્લિક કર્યા પછી ડ્રેગ કરવાથી એક કરતાં વધારે રો, કોલમ અથવા સેલ પસંદ કરી શકો છો.

ફોર્મ્યુલા અને ડેટા ઈનપુટ બાર (Formula and data input bar)

ટૂલબારની તરત નીચે અને વર્કશીટના ટોચ વિસ્તાર ઉપર એક મોટી ખાલી લીટી ચકાસો. તમે આ પહેલાના વિભાગમાં આકૃતિ 5.9માં આ તમે જોયું હશે. આ ખાલી જગ્યા વપરાશકર્તાને ડેટા અને સૂત્ર દાખલ કરવા માટે પૂરી પાડવામાં આવે છે. એક વાર ડેટા દાખલ કરવામાં આવે એટલે તે કોઈ ચોક્કસ સેલમાં દેખાશે.

જે સેલ હાલમાં પસંદ કરેલો હોય તેનું સરનામું ફોર્મ્યુલા બારમાં ડાબી બાજુએ જોવા મળે છે. જમણી બાજુએ $f(x)$ વડે સૂચવવામાં આવેલ બોક્સ એક વિસ્તાર પૂરો પાડે છે જેના વડે તમે સેલમાં ડેટા અથવા સૂત્રો દાખલ કરી શકો છો.

કાર્યવિસ્તાર (વર્કસ્પેસ - Workspace)

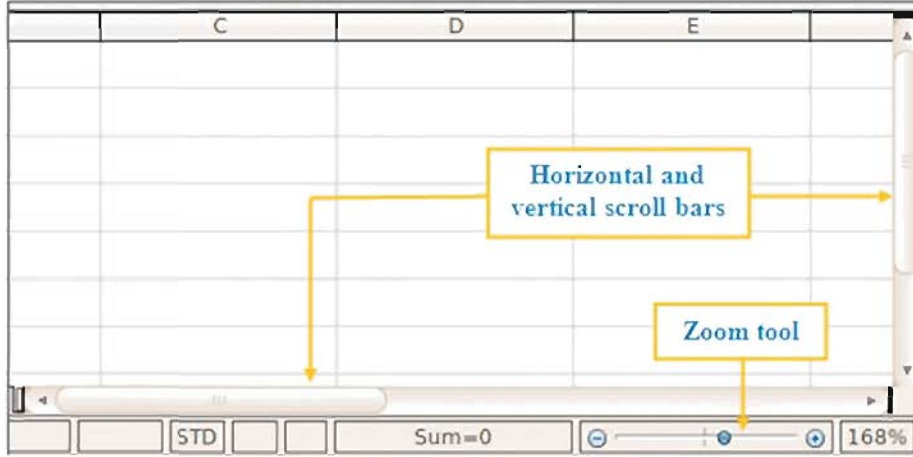
અનેક ખાલી સેલની જાળ (ગ્રીડ) એક વિસ્તાર બનાવે છે, જેમાં વપરાશકર્તાના ડેટા નજરે પડે છે. તમે કોઈ ચોક્કસ સેલ ઉપર ક્લિક કરી (તે સેલને કાર્યરત બનાવવા માટે) સીધા તેમાં ડેટા તેમજ સૂત્ર દાખલ કરી શકો છો. સેલમાં તમને સૂત્ર જોવા મળશે નહીં પણ ડેટા સ્વરૂપે તે સૂત્રનું પરિણામ જોવા મળશે.

ઝૂમ ટૂલ (Zoom tool)

વર્કશીટને ઝૂમ ઇન / આઉટ કરવા માટે ઝૂમ ટૂલ વપરાય છે. તમે વધારે સારા નિરીક્ષણ માટે વર્કશીટને ઝૂમ આઉટ કરી શકો. આકૃતિ 5.12માં ઝૂમ ટૂલને હાઈલાઈટ કર્યું છે જેના વડે જોઈ શકશો કે વર્કશીટ 168% ઝૂમ કરેલી છે.

સ્ક્રોલ બાર (Scroll bar)

કમ્પ્યુટર સ્ક્રીન(મોનિટર)ના કદ કરતાં વર્કશીટ મોટી હોવાથી આકૃતિ 5.12માં દર્શાવ્યા પ્રમાણે કેલ્સી આપોઆપ શીટની બંને બાજુએ તમને સ્ક્રોલબાર પૂરા પાડે છે. એટલે કે, સ્ક્રીન ઉપર તમે આડા અને ઊભા સ્ક્રોલબાર જોઈ શકશો. સ્ક્રોલબારના ઉપયોગથી તમને માહિતી નિહાળવાની વધારે અનુકૂળતા રહે છે, ખાસ કરીને જ્યારે ડેટા સ્ક્રીનમાં સમાતો ન હોય.



આકૃતિ 5.12 : ઝૂમ ટૂલ સાથે ઊભા અને આડા સ્ક્રોલબાર

સ્ટેટસ બાર (Status bar)

દસ્તાવેજની હાલની સ્થિતિ સાથે અન્ય માહિતી જેવી કે પસંદ કરેલા સેલનો સરવાળો, પાનાંની શૈલી, પસંદ કરવાની રીત અને સંગ્રહ ન કરેલાં પરિવર્તનો આ બાર દર્શાવે છે. આકૃતિ 5.13માં એક લાક્ષણિક સ્ટેટસ બાર દર્શાવેલ છે.



આકૃતિ 5.13 : સ્ટેટસ બાર

શીટ ટેબ (Sheet tab)

શીટ ટેબનો ઉપયોગ કરીને દસ્તાવેજની અલગ અલગ વર્કશીટ(ફક્ત શીટ પણ કહેવામાં આવે છે)માં તમે જઈ શકો છો. શીટ ટેબ વર્કશીટ માટે નેવિગેશન ટૂલ તરીકે કામ આપે છે. વર્તમાન વર્કશીટ ટેબનું નામ સફેદ રંગની પૃષ્ઠભૂમિ ધરાવે છે અને અન્ય બધી શીટ રાખોડી રંગની પૃષ્ઠભૂમિ ધરાવે છે. તમે વર્કશીટને પસંદ કરવા માટે જરૂરી વર્કશીટ ઉપર ફક્ત ક્લિક કરો. તમે "Sheet1" લખાણ ઉપર ફક્ત રાઈટ ક્લિક કરો, તે પછી "Rename sheet..." વિકલ્પ પસંદ કરો અને આપણને પસંદ નામ ટાઈપ કરીને પણ શીટનું નામ બદલી શકાય છે. આકૃતિ 5.14માં શીટ ટેબ દર્શાવેલ છે. આના વિકલ્પરૂપે, વિશિષ્ટ પરિસ્થિતિઓમાં, આકૃતિ 5.14માં દર્શાવ્યા પ્રમાણે બીજી વર્કશીટમાં જવા માટે તીરનો ઉપયોગ પણ કરી શકાય છે. તમે તીરનો ઉપયોગ કરીને હાલ પછીની વર્કશીટ (જમણી બાજુ), સૌથી અંતમાં (જમણી બાજુ), હાલ પહેલાંની વર્કશીટ (ડાબી બાજુ) અને પ્રથમ વર્કશીટ (ડાબી બાજુએથી સૌપ્રથમ) ઉપર જઈ શકો છો.

આકૃતિ 5.14 : વર્કશીટ પસંદ કરવા માટેની શીટ ટેબ

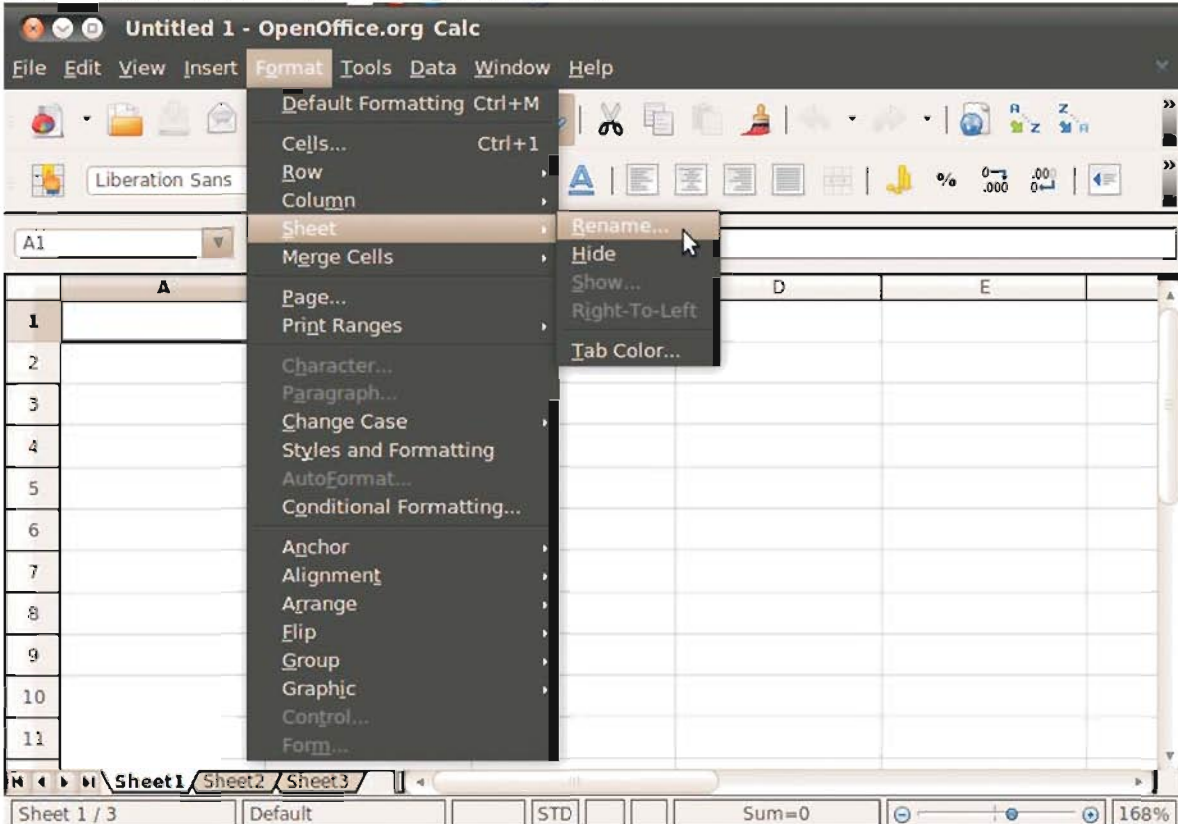
નવો કેલ્સી દસ્તાવેજ બનાવવો (Creating a Calc Document)

ચાલો, આપણે ખરીદીના બિલનો એક સાદો દસ્તાવેજ બનાવીએ. આ બિલમાં કંપનીનું નામ અને એડ્રેસ, ખરીદી કરેલી દરેક વસ્તુનાં નામ, કેટલા એકમ વસ્તુ ખરીદી, એકમ દીઠ ભાવ અને વસ્તુની કુલ રકમનો સમાવેશ કરવામાં આવે છે. આ બિલમાં દરેક વસ્તુની રકમનો કુલ સરવાળો, વેરાની રકમ, વળતર (જો કંઈ હોય તો) અને ગ્રાહકને આપવાની કુલ રકમનો પણ સમાવેશ કરવાની જરૂર છે. આ પ્રકારનો દસ્તાવેજ (ડોક્યુમેન્ટ) બનાવવા માટેનાં પગલાં નીચે જણાવેલાં છે :

પગલું 1 : સૌ પ્રથમ નીચે પ્રમાણે આદેશ આપો : Applications → Office → OpenOffice.org Spreadsheet આ આદેશ આપવાથી કેલ્સી ખૂલશે. તમે કેલ્સીનો સેતુ (ઇન્ટરફેઇસ) જોઈ શકશો.

પગલું 2 : તમે જોઈ શકશો કે કેલ્સી આપોઆપ ત્રણ વર્કશીટ દર્શાવે છે. જો કે અહીં અગાઉ જણાવ્યું તે પ્રમાણે ફક્ત રાઈટ ક્લિક કરી તમે જરૂરિયાત પ્રમાણે વર્કશીટ ડિલીટ કરી શકો છો તેમજ નવી વધારાની ઉમેરી પણ શકો છો. પૂર્વનિર્ધારિત આ ત્રણ વર્કશીટનાં નામ Sheet1, Sheet2, Sheet3 હોય છે. આમાંની કોઈ વર્કશીટને તમારી પસંદગીનું કોઈ ચોક્કસ નામ આપવા માટે નીચે જણાવ્યા પ્રમાણે કાર્ય કરો :

- સ્ટેટસ બારની તરત જ ઉપર સ્થિત વર્કશીટ ટેબ(આકૃતિ 5.14)માં જે વર્કશીટનું નામ બદલવાનું હોય તે વર્કશીટ ઉપર ક્લિક કરી તે વર્કશીટ પસંદ કરો.
- હવે આકૃતિ 5.15માં દર્શાવ્યા પ્રમાણે **Format** મેનૂ ઉપર ક્લિક કરો, તે પછી મેનૂનાં વિવિધ વિકલ્પોમાંથી **Sheet** વિકલ્પ પસંદ કરી અને તેનાં સબમેનૂના ઉપલબ્ધ વિકલ્પોમાંથી **Rename** પસંદ કરો.
- હવે યોગ્ય નામ (જેમ કે "bill") ટાઈપ કરો અને કામગીરી પૂર્ણ કરવા માટે **OK** બટન ઉપર ક્લિક કરો. હવે વર્કશીટના નામમાં થયેલા ફેરફારનું નિરીક્ષણ કરો.



આકૃતિ 5.15 : શીટનું નામ બદલવું

પગલું 3 : હવે કોષ્ટક 5.3માં આપેલી માહિતી 'bill' નામની વર્કશીટમાં દાખલ કરો.

Sr. No.	Item	Quantity	Unit price
1	Pen	5	10
2	Pencil	6	2
3	Pencil box	1	50
4	Notebook	10	20
5	Notebook cover	10	2

કોષ્ટક 5.3 : ખરીદીના બિલની માહિતી

'bill' નામની વર્કશીટમાં ડેટા દાખલ કરવા માટે નીચે મુજબ કાર્ય કરો :

- સેલ **A1** ઉપર લેફ્ટ ક્લિક કરો. હવે શબ્દ **"Sr. No."** દાખલ કરો અને એન્ટર કી દબાવો.
- સેલ **B1** ઉપર લેફ્ટ ક્લિક કરો. હવે શબ્દ **"Item"** દાખલ કરો અને એન્ટર કી દબાવો.
- સેલ **C1** ઉપર લેફ્ટ ક્લિક કરો. હવે શબ્દ **"Quantity"** દાખલ કરો અને એન્ટર કી દબાવો.
- સેલ **D1** ઉપર લેફ્ટ ક્લિક કરો. હવે શબ્દ **"Unit price"** દાખલ કરો અને એન્ટર કી દબાવો.

તમે ઊભી હરોળ(કોલમ)નું કદ બદલવા માટે કોલમને ડ્રેગ કરી શકો. તમે જ્યારે વસ્તુઓનાં નામ દાખલ કરી રહ્યા હોય ત્યારે તે ઊભી હરોળનું માપ બદલવાની જરૂર પડી શકે છે. આ કાર્ય કરવા માટે, જે ઊભી હરોળનું માપ તમે બદલવા ઇચ્છો છો તે ઊભી હરોળની ધાર પર માઉસ મૂકો અને જરૂરી લંબાઈ સુધી તેને ડ્રેગ કરો (ખેંચો). તમે જોઈ શકો છો કે પ્રથમ શીર્ષક લીટી તૈયાર થઈ ગઈ છે. તમે એક અથવા વધારે સેલ પસંદ કરી ફોર્મેટિંગ ટૂલબાર વડે તેને ઘાટા (bold) કરી શકો છો.

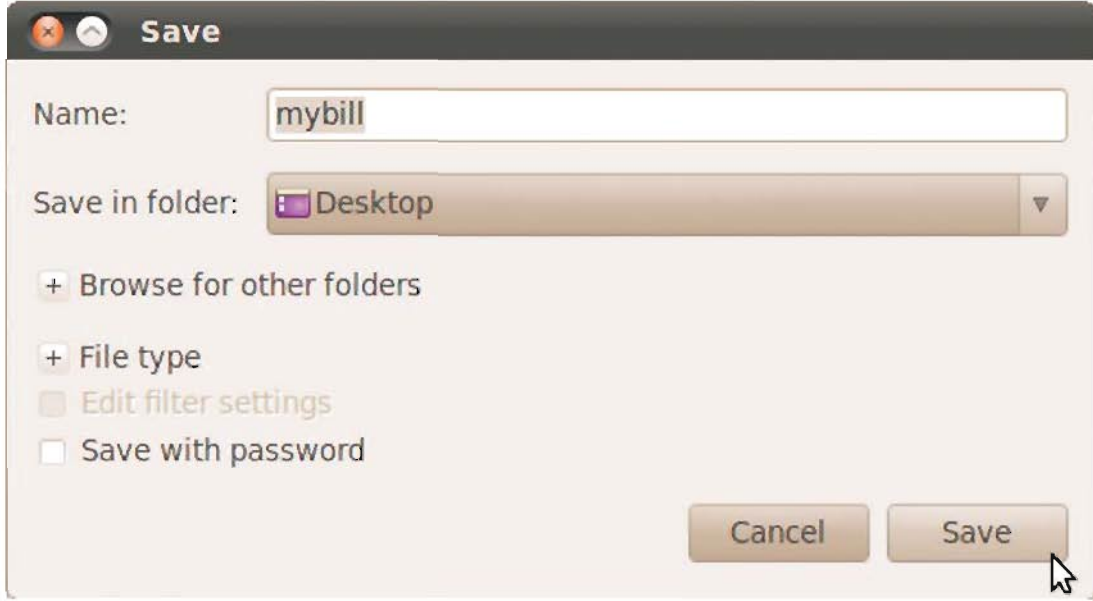
હવે સેલ A2 માં સંખ્યા "1", સેલ B2માં શબ્દ "Pen", સેલ C2માં સંખ્યા "5" અને સેલ D2માં સંખ્યા "10" દાખલ કરો. આ જ રીતે બાકીની લીટીઓમાં માહિતી દાખલ કરો. હવે વર્કશીટ આકૃતિ 5.16માં દર્શાવ્યા પ્રમાણે દેખાશે.

A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	
2	1	Pen	5	10	
3	2	Pencil	6	2	
4	3	Pencil box	1	50	
5	4	Notebook	10	20	
6	5	Notebook cover	10	2	
7					
8					
9					
10					
11					
12					
13					
14					

આકૃતિ 5.16 : bill વર્કશીટમાં દાખલ કરેલી માહિતી

પગલું 4 : હવે યોગ્ય નામ સાથે વર્કશીટનો સંગ્રહ કરો (સેવ કરો) જેથી દાખલ કરેલી માહિતી આકસ્મિક ગુમાવવી ન પડે. વર્કશીટનો સંગ્રહ કરવા માટે નીચે પ્રમાણે આદેશ આપો.

- **File → Save** ક્લિક કરો. આથી આકૃતિ 5.17માં દર્શાવ્યા પ્રમાણે એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર દેખાશે; તેમાં ફાઈલનું નામ અને પાથ (જ્યાં તમે ફાઈલનો સંગ્રહ કરવા ઇચ્છો છો) જણાવો.
- હવે Save બટન ઉપર ક્લિક કરો.



આકૃતિ 5.17 : કેલ્સી દસ્તાવેજનો સંગ્રહ કરવો

સ્પ્રેડશીટનો **OpenOffice.org Calc** ફાઈલ સ્વરૂપે સંગ્રહ કરે છે અને તેનું અનુલંબન (એક્સટેન્શન) .ods હોય છે. તમે ફાઈલનો જે જગ્યાએ સંગ્રહ કરો તે સ્થાન તમારે યાદ રાખવું પડે જેથી ભવિષ્યમાં તે ફાઈલ મેળવી શકો.

હજુ સુધી આપણે ફક્ત જરૂરી ડેટા જ દાખલ કર્યો છે અને તે ફાઈલનો સંગ્રહ કર્યો છે. પણ આપણે દરેક વસ્તુની કિંમત, બધી વસ્તુઓની કુલ કિંમત, કર અને ચોખ્ખી ચૂકવવાપાત્ર રકમની ગણતરી કરી નથી. કોઈ ચોક્કસ વસ્તુ- (જેમ કે પેન)ની કિંમતની ગણતરી માટે આપણે વસ્તુની કિંમત(એક પેનની કિંમત)ના ખરીદેલ જથ્થા (પેનની સંખ્યા) સાથે ગુણાકાર કરવો પડે. પ્રથમ આડી હરોળ પ્રમાણે નંગદીઠ ₹ 10ના ભાવે 5 પેનની ખરીદી કરી હતી. આથી પહેલી વસ્તુની કુલ કિંમત ₹ 50 થાય. આ જ રીતે અન્ય વસ્તુઓ માટે ચૂકવવાપાત્ર રકમ નીચે પ્રમાણે છે :

પ્રથમ આડી હરોળ → 5 પેન, દરેકના ₹ 10	= 5*10 ₹	= ₹ 50
બીજી આડી હરોળ → 6 પેન્સિલ, દરેકના ₹ 2	= 6*2 ₹	= ₹ 12
ત્રીજી આડી હરોળ → 1 પેન્સિલ બોક્સ, દરેકના ₹ 50	= 1*50 ₹	= ₹ 50
ચોથી આડી હરોળ → 10 નોટબુક, દરેકના ₹ 20	= 10*20 ₹	= ₹ 200
પાંચમી આડી હરોળ → 10 નોટબુક કવર, દરેકના ₹ 2	= 10*2 ₹	= ₹ 20

ઉપરની ગણતરી પ્રમાણે, આપણે એક સામાન્ય નિયમ તારવી શકીએ કે જો આપણે વસ્તુનાં x એકમ ખરીદીએ, દરેક એકમની y કિંમત હોય તો ચૂકવવાપાત્ર રકમ $x*y$ થાય. આ રકમ (પરિણામ) "**Amount**" નામની છેલ્લી ઊભી હરોળમાં દાખલ કરવા માટે આપણે સંખ્યાનો ગુણાકાર જાતે કરીને સેલમાં લખી શકીએ. જો કે તેનાથી સારો વિકલ્પ આપણે જાતિગત સૂત્ર (generic formula) લખીએ તે છે.

સેલમાં કોઈ સૂત્ર દાખલ કરવા માટે આપણે સીધો ડેટા (જેમ કે $=5*10$) દાખલ કરી શકીએ અથવા ડેટા જે સેલમાં ઉપલબ્ધ છે તે સેલનો નિર્દેશ પણ આપણે કરી શકીએ. એટલે કે, સેલ C2 માં જો 5 હોય અને સેલ D2માં 10 હોય તો આ દાખલામાં ($=5*10$)ની કિંમત ($=C2*D2$) બરાબર થાય. કોઈ પણ સૂત્રની શરૂઆતમાં બરાબરનું ચિહ્ન '=' લખવું જરૂરી છે અને તેમ કરવામાં નિષ્ફળ જતાં સેલમાં નિયમ પ્રમાણે શાબ્દિક માહિતી તરીકે દાખલ થાય છે, (સંખ્યા દાખલ કરી હોય તો પણ) અને તેના ઉપર ગણતરીઓ કરવામાં આવતી નથી.

સીધી કિંમત કોઈ સેલમાં દાખલ કરવા કરતાં સેલ નિર્દેશ(cell reference)ના ઉપયોગના કેટલાક લાભ છે. પહેલો લાભ એ છે કે આપણે જાતે અંકગણિતની ક્રિયા કરવાની જરૂર પડતી નથી; કે જે અમુક સમયે ભૂલમાં પણ પરિણમી શકે. બીજો લાભ એ છે કે જ્યારે આપણે સેલમાં કિંમતને બદલીએ ત્યારે સૂત્રનું પરિણામ પણ આપોઆપ બદલાઈ જાય છે. ત્રીજો લાભ એ છે કે જ્યારે ડેટાનો કોઈ જથ્થો (બ્લોક) એક જગ્યાએથી બીજી જગ્યાએ ખસેડવામાં આવે અથવા કેટલીક ઊભી કે આડી હરોળ ઉમેરવામાં / કાઢી નાખવામાં આવે તે સમયે સૂત્રમાં રહેલાં સેલ નિર્દેશ આપોઆપ બદલાઈ જશે.

ચાલો, આપણે અગાઉની 'bill' નામની વર્કશીટમાં દરેક વસ્તુદીઠ ચૂકવવાપાત્ર રકમની ગણતરી માટે એક ઊભી હરોળ વર્કશીટમાં ઉમેરીએ. હવે નીચે પ્રમાણે કાર્ય કરો :

- સેલ E1 પસંદ કરો અને તેમાં **"Amount"** શબ્દ દાખલ કરો. તે પછી એન્ટર કી દબાવો. તમે આ શબ્દને ઘાટા (બોલ્ડ) કરી શકો છો.
- આકૃતિ 5.18માં દર્શાવ્યા પ્રમાણે સેલ E2 માં $=C2*D2$ ટાઈપ કરી સેલમાં સૂત્ર ઉમેરી શકો છો.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10

SUM f(x) X ✓ =c2*d2

	A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount	
2	1	Pen	5	10	=c2*d2	
3	2	Pencil	6	2		
4	3	Pencil box	1	50		
5	4	Notebook	10	20		
6	5	Notebook cover	10	2		
7						
8						
9						
10						
11						
12						
13						

bill/Sheet2/Sheet3

Sheet 1 / 3 Default INSRT STD Sum=0 128%

આકૃતિ 5.18 : સરળ સૂત્ર ઉમેરવું

- સેલ E2માં સૂત્ર દાખલ કર્યા પછી તમે જ્યારે એન્ટર કી દબાવો છો ત્યારે તમે સેલમાં ગુણાકારનું પરિણામ જોઈ શકશો.
- તમે સેલ E3માં ($=C3*D3$), સેલ E4માં ($=C4*D4$), સેલ E5માં ($=C5*D5$) અને સેલ E6માં ($=C6*D6$) સૂત્ર લખી શકો છો.
- આનો વિકલ્પ સેલ E2ની માહિતી તમે ઊભી હરોળના બાકીના સેલમાં સેલ E6 સુધી ફક્ત ડ્રેગ કરવાનો છે. આથી સૂત્રની નકલ આપોઆપ થઈ જશે. સેલની માહિતી ડ્રેગ કરવા માટે, સેલ ઉપર ક્લિક કરો. તે પછી તે સેલના જમણી બાજુના નીચેના ખૂણાને જરૂરી સંખ્યાના સેલ સુધી ડ્રેગ કરો.
- આકૃતિ 5.19માં દર્શાવ્યા પ્રમાણેનો આઉટપુટ તમે જોઈ શકશો.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10

E2:E6 f(x) Σ = =C2*D2

	A	B	C	D	E	F
	Sr. No.	Item	Quantity	Unit price	Amount	
1	1	Pen	5	10	50	
2	2	Pencil	6	2	12	
3	3	Pencil box	1	50	50	
4	4	Notebook	10	20	200	
5	5	Notebook cover	10	2	20	
6						
7						
8						
9						
10						
11						
12						
13						

Sheet 1 / 3 Default STD Sum=332 128%

આકૃતિ 5.19 : એક સેલની કિંમતની ગણતરી કર્યા પછી અન્ય સેલમાં ફેરવું

આપણે જુદી જુદી વસ્તુઓની ચુકવવાપાત્ર રકમનો કુલ સરવાળો કરવાની પણ જરૂર છે. 'Amount' હરોળની બધી સંખ્યાઓનો સરવાળો કરવા માટે SUM વિધેયનો ઉપયોગ કર્યો છે. SUM વિધેયનું બટન સ્ક્રીનના ઉપરના ભાગમાં ફંક્શન બાર(function bar)માં આવેલું હોય છે. SUM વિધેય બટનનો ઉપયોગ કરી તમે પસંદ કરેલી સેલ રેન્જમાં રહેલી સંખ્યાનો સરવાળો આપોઆપ કરી શકો છો. આકૃતિ 5.20માં SUM વિધેય દર્શાવેલું છે.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10

SUM f(x) X ✓ =sum(E2,E6)

	A	B	C	D	E	F
	Sr. No.	Item	Quantity	Unit price	Amount	
1	1	Pen	5	10	50	
2	2	Pencil	6	2	12	
3	3	Pencil box	1	50	50	
4	4	Notebook	10	20	200	
5	5	Notebook cover	10	2	20	
6						
7						
8						
9						
10						
11						
12						
13						

Sheet 1 / 3 Default INSRT STD Sum=0 128%

આકૃતિ 5.20 : SUM વિધેય

આકૃતિ 5.21માં દર્શાવ્યા પ્રમાણે સ્ક્રીનમાં સૌથી ઉપર ફોર્મ્યુલા બારમાં ઉપલબ્ધ ચિહ્ન (Σ તરીકે સૂચિત) પણ વાપરી શકાય છે.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Libertion Sans 10

E7 Σ =

	A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount	
2	1	Pen	5	10	50	
3	2	Pencil	6	2	12	
4	3	Pencil box	1	50	50	
5	4	Notebook	10	20	200	
6	5	Notebook cover	10	2	20	
7						
8						
9						
10						
11						
12						
13						

bill Sheet2 Sheet3

Sheet 1 / 3 Default STD Sum=0 128%

આકૃતિ 5.21 : ઓટોસમ (autosum) ચિહ્નનો ઉપયોગ

સ્પ્રેડશીટમાં સેલની રેન્જ (range) દર્શાવવા માટે શરૂઆતના સેલનું એડ્રેસ અને છેલ્લા સેલનું એડ્રેસને વિસર્ગ(:)થી છૂટા પાડીને દર્શાવવામાં આવે છે. દા.ત., E2:E6 એ E2થી E6 સેલનો નિર્દેશ કરે છે. આના વિકલ્પરૂપે, તમે સેલ E7 પસંદ કરી તેમાં **=E2+E3+E4+E5+E6** સૂત્ર ટાઇપ પણ કરી શકો છો. સેલ D7 પસંદ કરી તેમાં શીર્ષક "Total" લખીને એન્ટર કી દબાવો. આથી તમે આકૃતિ 5.22માં દર્શાવ્યા પ્રમાણે સ્ક્રીન જોઈ શકશો.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Libertion Sans 10

D7 Σ = Total

	A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount	
2	1	Pen	5	10	50	
3	2	Pencil	6	2	12	
4	3	Pencil box	1	50	50	
5	4	Notebook	10	20	200	
6	5	Notebook cover	10	2	20	
7				Total	332	
8						
9						
10						
11						
12						
13						

bill Sheet2 Sheet3

Sheet 1 / 3 Default STD Sum=0 128%

આકૃતિ 5.22 : કુલ કિંમતની ગણતરી

ચાલો, હવે આપણે ઉપર ગણતરી કરેલી કુલ કિંમત ઉપર કરની ગણતરી કરીએ. જો આપણે કુલ બિલની કિંમત ઉપર 5% કર ગણીએ તો કર રકમનું સૂત્ર મેળવવા માટે નીચે પ્રમાણે કરો :

$$\text{કર} = \text{કુલ રકમ} \times 5/100$$

- સેલ D8 પસંદ કરો અને તેમાં **"Taxes"** શીર્ષક લખો.
- સેલ E8 પસંદ કરો અને તેમાં $=E7*5/100$ સૂત્ર લખો.

ગ્રાહકને કુલ ચૂકવવાપાત્ર રકમની ગણતરી માટે કુલ રકમમાં કર ઉમેરવો પડે. સેલ E7 અને સેલ E8ની સંખ્યાનો સરવાળો કરી સેલ E9માં પ્રદર્શિત કરો. આપણે ઉપર ચર્ચા કરી તે પ્રમાણે sum વિધેયનો તમે ઉપયોગ કરી શકો અથવા સેલ E9 પસંદ કરી તેમાં $(=E7+E8)$ સૂત્ર સીધું લખી શકો.

સેલ D9 પસંદ કરો અને તેમાં શીર્ષક **"Net Amount"** લખી એન્ટર કી દબાવો. તમે આકૃતિ 5.23માં દર્શાવ્યા પ્રમાણે પરિણામ જોઈ શકશો.

	A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount	
2	1	Pen	5	10	50	
3	2	Pencil	6	2	12	
4	3	Pencil box	1	50	50	
5	4	Notebook	10	20	200	
6	5	Notebook cover	10	2	20	
7				Total	332	
8				Taxes	16.6	
9				Net Amount	348.6	
10						
11						
12						
13						
14						

આકૃતિ 5.23 : ચૂકવવાપાત્ર રકમની ગણતરી

તમે અહીં જોઈ શકશો કે કુલ ચૂકવવાપાત્ર રકમ 348.6 થાય છે. તેને વધારે પરિચિત બનાવવા માટે આકૃતિ 5.23માં દર્શાવ્યા પ્રમાણે ટૂલબાર ઉપર ક્લિક કરીને આપણે દશાંશ ચિહ્ન ઉમેરી શકીએ છીએ. એક દશાંશ ચિહ્ન ઉમેરવાથી કુલ રકમની કિંમત 348.60 બનશે.

ધારો કે તમે વર્કશીટની સઘળી માહિતી એવી રીતે ખસેડવા ઇચ્છો છે કે જેથી તમે કંપનીનું નામ અને તારીખનો સમાવેશ કરી શકો. આ માટે નીચે જણાવેલાં પગલાંને અનુસરો :

- માઉસનો ઉપયોગ કરીને બધી માહિતીનો સમાવેશ થાય તે પ્રમાણે બધા સેલને પસંદ કરો. ઉપરના ઉદાહરણમાં, સંપૂર્ણ માહિતીનો સમાવેશ કરતા સેલ A1થી E9 છે.
- એડિટ (edit) મેનૂ પસંદ કરી તેમાંથી **"cut"** આદેશ પસંદ કરી આપણે પસંદ કરેલો ડેટા કટ કરો.
- હવે સેલ B3 ઉપર જાઓ અને ત્યાં ડેટા પેસ્ટ કરો.
- નિશ્ચિત કરો કે તમારા ડેટામાં કોઈ ફેરફાર થયેલો નથી.

તમે સૂત્રોમાં પરિવર્તન ઉપર ધ્યાન આપ્યું હશે. કુલ રકમ, કર અને ચૂકવવાપાત્ર રકમ હવે E હરોળને બદલે F હરોળનો નિર્દેશ કરે છે. સેલ જો એક જગ્યાએથી બીજી જગ્યાએ ખસેડવામાં આવે તો કેલ્સી સ્વયં રીતે સેલ નિર્દેશો બદલી નાખે છે, સિવાય કે સ્પષ્ટપણે તેમ ન કરવા સૂચના આપી હોય. હવે પછીના પ્રકરણમાં આપણે સાપેક્ષ સ્થાનાંક (relative address) અને નિરપેક્ષ સ્થાનાંક (absolute address) જેવી કાર્યરીતિ ઉપર વધારે શીખીશું.

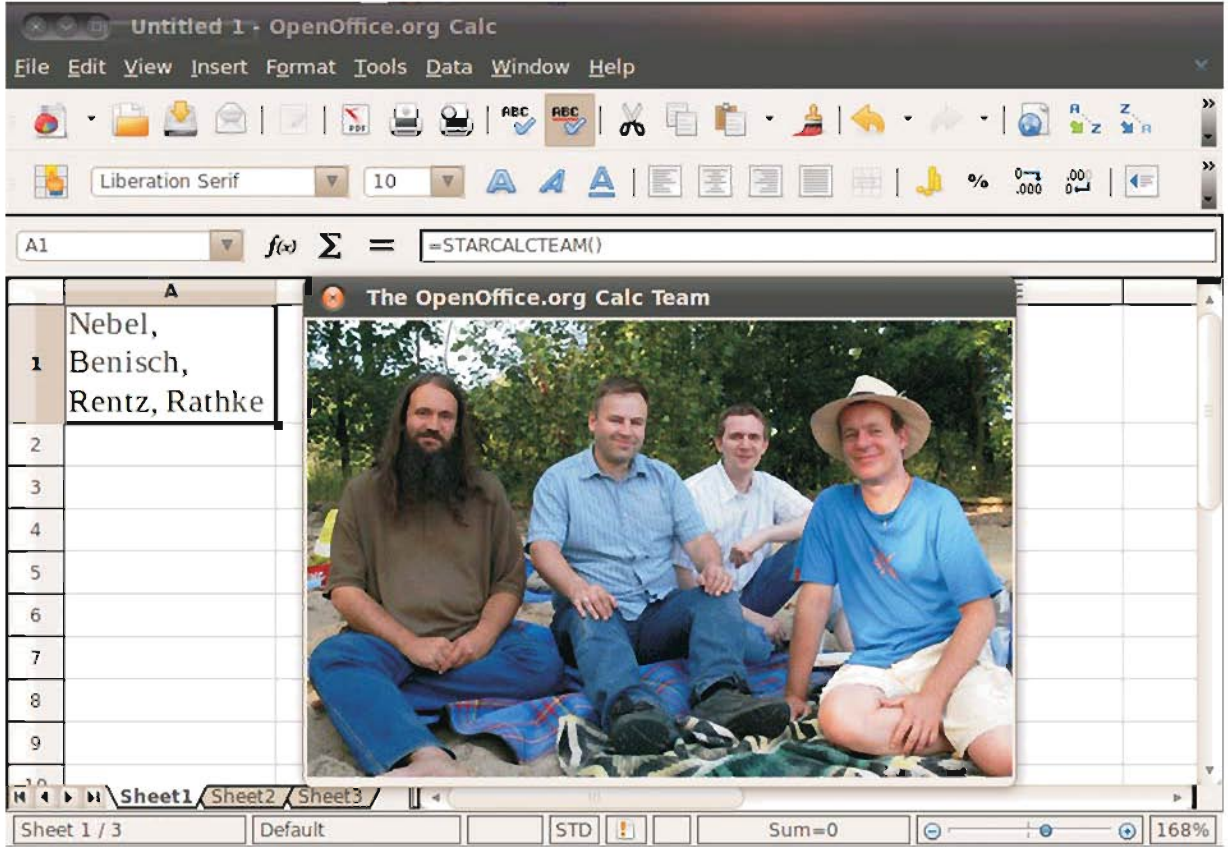
ફાઈલનો સંગ્રહ કરવો અને ફરી ખોલવી (Saving and Re-opening File)

સ્પ્રેડશીટમાં (અથવા કોઈ પણ કમ્પ્યુટર વિનિયોગમાં) કામ કરતી વખતે તમારા કામનો વારંવાર સંગ્રહ કરવાની જરૂરિયાત હોય છે. વીજળીમાં વધઘટ થવી કે આવી અન્ય કોઈ સમસ્યાને કારણે તમે દાખલ કરેલો ડેટા ગુમાવવાનો સમય પણ આવે. ફાઈલનો વારંવાર સંગ્રહ કરવાની આદત સારી છે અને આકસ્મિક રીતે ડેટા ગુમાવવાને અટકાવે છે.

ફાઈલ ઉપરનું કામ એક વખત પૂર્ણ થઈ જાય પછી તમારે ફાઈલ બંધ કરવી પડે. તમે વિન્ડોમાં સોથી ઉપર રહેલા ક્લોઝ (close) બટન ઉપર ક્લિક કરી શકો અથવા File મેનૂમાંથી close આદેશ આપો. તે ફાઈલ ફરી ખોલવા માટે તમારે File મેનૂમાંથી Open આદેશ આપવો પડે. આ કાર્ય માટે તમે સ્ટાન્ડર્ડ ટૂલનો ઉપયોગ પણ કરી શકો છો.

કેલ્સી વડે ઓપન ઓફિસ સ્યૂટ બનાવનારી ટુકડીને મળો (Meet the Developers of the OpenOffice Suit via Calc)

હવે તમે ઓપનઓફિસમાં સ્પ્રેડશીટ કઈ રીતે કાર્ય કરે છે તે જાણો છો. નીચે જણાવ્યા પ્રમાણે કાર્ય કરો. કેલ્સીમાં એક નવી વર્કબુક ખોલો. કોઈ પણ સેલ પસંદ કરી તેમાં =starcalc() ટાઈપ કરો, આથી તે ઓપનઓફિસ સ્યૂટ બનાવનારી ટુકડીના સભ્યોનો ફોટોગ્રાફ પ્રદર્શિત કરશે, આકૃતિ 5.24માં કેલ્સી બનાવનારી ટુકડી પ્રદર્શિત કરે છે.



આકૃતિ 5.24 : કેલ્સી બનાવનારી ટુકડી

સારાંશ

આપણે આ પ્રકરણમાં સ્પ્રેડશીટ શું છે તે બાબત શીખ્યા અને આવા પેકેજનો ઉપયોગ કરીને બનાવવામાં આવતા કેટલાક લાક્ષણિક વિનિયોગો વિશે જાણ્યું. આ પ્રકરણમાં ઓપન ઓફિસ સ્યૂટના સ્પ્રેડશીટ પેકેજના પરિચય અને મૂળભૂત કામગીરી ઉપર ધ્યાન કેન્દ્રિત કર્યું છે. તેની લાક્ષણિકતાઓ અને કામગીરી બાબતની ચર્ચા વાસ્તવિક ઉદાહરણને કમબંધ રીતે સમજાવીને કરેલી છે. તમે આ પાઠમાં જે મૂળભૂત બાબતો શીખ્યા છો, તેના વડે તમે નવી વર્કશીટ બનાવી શકો, તેમાંના ડેટામાં સુધારા-વધારા કરી શકો, કેટલાંક મૂળભૂત સૂત્રો વડે ડેટા ઉપર પ્રક્રિયા તેમજ વિશ્લેષણ કરી શકો, સ્પ્રેડશીટનો સંગ્રહ કરી શકો અને જરૂર હોય ત્યારે ફરી ખોલી શકો છો.

સ્વાધ્યાય

1. સ્પ્રેડશીટ પેકેજનો ઉપયોગ કરી તૈયાર થતા યોગ્ય વિનિયોગની યાદી બનાવો.
2. કેલ્સીમાં ફાઈલનો સંગ્રહ કરવાની અને ફરી ખોલવાની ક્રિયા સમજાવો.
3. કેલ્સી વર્કશીટમાં સૂત્રો કઈ રીતે કાર્ય કરે છે તે સમજાવો.
4. કેલ્સીની હયાત શીટનું નામ આપણે બદલી શકીએ? કઈ રીતે?
5. જો કેલ્સી દસ્તાવેજમાં સ્ટાન્ડર્ડ ટૂલબાર દૃશ્યમાન ન હોય તો તમે શું કરશો?
6. નીચેનામાંથી સાચો વિકલ્પ પસંદ કરો :
 - (1) કેલ્સી નીચેનામાંથી કયા પ્રકારના પેકેજનો નિર્દેશ કરે છે?
 - (a) સ્પ્રેડશીટ
 - (b) મલ્ટીશીટ
 - (c) ડબલ શીટ
 - (d) નક્કી કરી શકાતું નથી.
 - (2) નીચેનામાંથી કયો વિનિયોગ કેલ્સી માટે યોગ્ય નથી?
 - (a) સરવૈધું તૈયાર કરવું
 - (b) પરિણામનું વિશ્લેષણ
 - (c) કોઈ ઉત્પાદન વિશે માહિતી રજૂ કરવી
 - (d) આપેલા તમામ
 - (3) કેલ્સીમાં બનાવેલી વર્કશીટનું અનુલંબન નીચેનામાંથી શું હોય છે?
 - (a) .ods
 - (b) .odd
 - (c) .xls
 - (d) .obj
 - (4) કેલ્સીનાં કોઈ સેલમાં જો =starcalcteam() દાખલ કરવામાં આવે તો નીચેનામાંથી શું વર્કશીટમાં જોવા મળશે?
 - (a) તારાઓ
 - (b) કેલ્સી બનાવનાર ટુકડીની તસવીર
 - (c) કેલ્સી પરવાના (લાયસન્સ) માહિતી
 - (d) કેલ્સી આવૃત્તિ માહિતી
 - (5) કેલ્સી દસ્તાવેજની વર્કશીટમાં કોઈ વ્યક્તિએ દાખલ કરેલી કિંમતોના સરવાળાની ગણતરી કેવી રીતે કરી શકે?
 - (a) જાતે ગણતરી કરીને
 - (b) autotsum વડે
 - (c) સૂત્ર વડે
 - (d) ઉપરોક્ત તમામ
 - (6) જો આપણે વર્કશીટના અન્ય કોઈ સેલનો નિર્દેશ ધરાવતા સૂત્રવાળા સેલને અન્ય જગ્યાએ ખસેડીએ તો સૂત્રમાંના સેલનું શું થશે?
 - (a) ગંતવ્ય પર સેલની રો (આડી હરોળ) અને કોલમ (ઊભી હરોળ) બદલાઈ જશે.
 - (b) ગંતવ્ય પર સેલના રો નંબર બદલાઈ જશે.
 - (c) ગંતવ્ય પર સેલના કોલમ નંબર બદલાઈ જશે.
 - (d) કશું બદલાશે નહીં.

પ્રાયોગિક સ્વાધ્યાય

1. આ પ્રકરણમાં ખરીદીના બિલના ઉદાહરણની ચર્ચા કરી હતી તેની રચના કરો.
2. તમે કોઈ સુપર સ્ટોરમાંથી મેળવેલ બિલનો અભ્યાસ કરો અને કેલ્સીમાં તેની રચના કરો.
3. તમારા છ જુદા જુદા વિષયોના ગુણનો એક કેલ્સી દસ્તાવેજ બનાવો. તે પછી ડેટાના આધારે કુલ ગુણ, સરેરાશ અને ટકા શોધો.
4. તમારા મિત્રોના ગુણ ઉમેરી આ સ્વાધ્યાયના ત્રીજા ઉદાહરણને વિસ્તારો.



કેલ્સીમાં ડેટાનું એડિટિંગ અને ફોર્મેટિંગ

અગાઉના પ્રકરણમાં ચર્ચા કરી હતી તે પ્રમાણે સ્પ્રેડશીટમાં માહિતીનો સંગ્રહ કરવા માટેનો મૂળભૂત એકમ સેલ છે. ઊભી હરોળ (કોલમ) અને આડી હરોળ (રો) એકબીજાને છેદવાથી બનતા ઘણા બધા સેલનો સમૂહ એટલે જ સ્પ્રેડશીટ. બધો જ ડેટા, સૂત્રો અને વિષયો આ સેલની અંદર લખવામાં આવે છે. આ પ્રકરણ તમને ડેટા એડિટિંગ (ડેટામાં સુધારા-વધારા કરવાની ક્રિયા) અને ફોર્મેટિંગ (ડેટાને યોગ્ય અને આકર્ષક સ્વરૂપમાં ગોઠવવા) સંબંધિત મૂળભૂત કામગીરી વિશે સંક્ષિપ્ત રૂપરેખા આપશે.

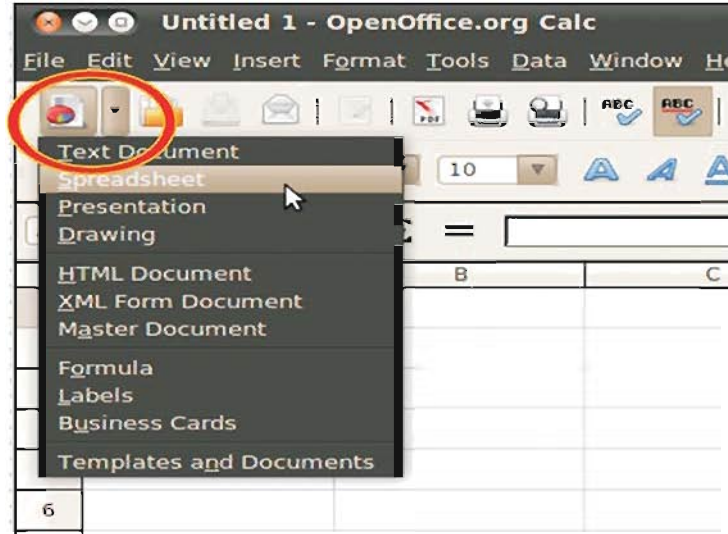
કેલ્સીમાં મૂળભૂત વર્કશીટ કાર્યો (Basic worksheet operations in Calc)

કેટલાંક કાર્યો જેવાં કે નવી અથવા હયાત સ્પ્રેડશીટ ખોલવી, સ્પ્રેડશીટનો સંગ્રહ કરવો, save as આદેશ વડે આખી સ્પ્રેડશીટનું નામ બદલવું, વર્કશીટનું નામ બદલવું, અનેક વર્કશીટ ઉમેરવી / દૂર કરવી વગેરે સ્પ્રેડશીટ અથવા વર્કશીટ સ્તર ઉપર વારંવાર થતી પ્રક્રિયાઓ છે. આ વિભાગમાં આ માટેના આદેશોની ચર્ચા કરી છે.

નવો દસ્તાવેજ બનાવવો (Creating a new document)

જ્યારે તમે OpenOffice.org Calc વિનિયોગ ખોલો છો ત્યારે તે આપોઆપ એક નવો (ખાલી) દસ્તાવેજ ખોલશે. જો તમે પહેલાંથી જ આ દસ્તાવેજનો ઉપયોગ કરી લીધો હોય અને તેમાં તમે કેટલીક માહિતી લખી હોય તો તમે બીજો નવો સ્પ્રેડશીટ દસ્તાવેજ ખોલી શકો છો. નવો કેલ્સી દસ્તાવેજ ખોલવા માટે, એટલે કે કેલ્સી સ્પ્રેડશીટ ખોલવા માટે નીચે મુજબની ક્રિયાઓ કરો :

- મેનૂબાર ઉપરથી **File → New → Spreadsheet** પસંદ કરો
- CTRL + N દબાવો, અથવા
- સ્ક્રીનના ઉપરના ભાગમાં ટૂલબાર ઉપરથી **New Document** આઈકોન ઉપર ક્લિક કરો અને સ્પ્રેડશીટ પસંદ કરો. આઈકોન (લંબગોળમાં દર્શાવેલ) આકૃતિ 6.1માં બતાવ્યા પ્રમાણે દેખાશે.



આકૃતિ 6.1 : નવી સ્પ્રેડશીટ ખોલવા માટેનો આઈકોન (icon)

જ્યારે તમે કોઈ નવી સ્પ્રેડશીટ ખોલો છો ત્યારે તે તમને ત્રણ અલગ અલગ વર્કશીટ આપે છે; જે રીતે તમે ફક્ત ત્રણ પાનાંની ગણિત માટે નોટબુક ખરીદી હોય જેમાં આડી હરોળો અને ઊભી હરોળોનું જાળું (ગ્રીડ) હોય છે. તમે પ્રકરણ 5માં ખરીદીના બિલની માહિતી દાખલ કરવા માટે કઈ રીતે ત્રણ વર્કશીટ સાથે સ્પ્રેડશીટ ખોલી શકાય તે શીખ્યા છો.

કોઈ હયાત કેલ્સી દસ્તાવેજને ઘણી બધી રીતે ખોલી શકાય. આમાંની પહેલી રીતમાં સૌપ્રથમ કેલ્સી ખોલી ફાઈલ મેનૂમાંથી open આદેશ આપો. આ માટેના આદેશોની શ્રેણી નીચે આપેલી છે :

- પસંદ કરો: **File → Open**
- **Open** ડાયલોગ બોક્સમાં પૂર્વનિર્ધારિત ફોલ્ડરમાંથી કે અન્ય ચોક્કસ ફોલ્ડરમાંથી તમારી જરૂરી ફાઈલ પસંદ કરો.
- આ ક્રિયા **Open** બટન ઉપર ક્લિક કરી પૂર્ણ કરો.

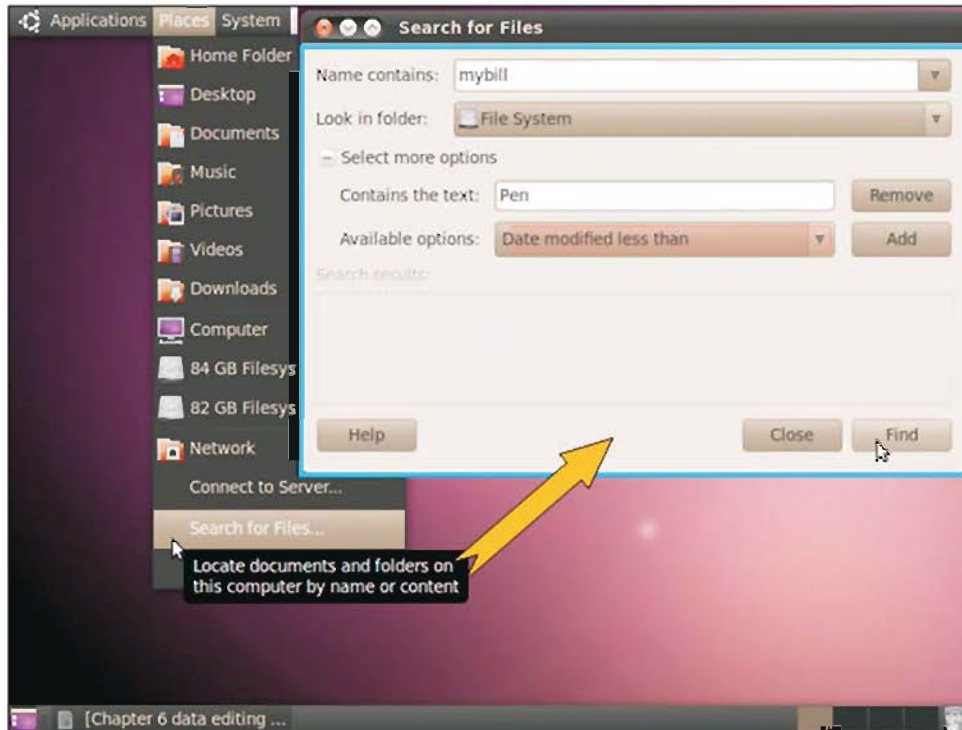
અન્ય વિકલ્પમાં, પહેલાં તમે કમ્પ્યુટરમાં જરૂરી ફાઈલ શોધી કાઢો પછી નીચે જણાવ્યા પ્રમાણે કેલ્સી સોફ્ટવેરમાં તેને ખોલો.

- **Places → Documents** આદેશ આપીને **Documents** ફોલ્ડર ખોલો.
- તમારી જરૂરી એક્સીટ ફાઈલ પસંદ કરી તેને ખોલો.

જો ઉપર જણાવેલાં પગલાંને અનુસરીને પણ તમે તમારી ફાઈલ શોધી ન શકો તો ઉબુન્ટુની સર્ચ (search) યુટિલિટી વડે ફાઈલને શોધી શકો છો. ખાસ કરીને જ્યારે તમારી જરૂરી ફાઈલની જગ્યા તમને યાદ ન હોય ત્યારે આ યુટિલિટી ખૂબ ઉપયોગી છે.

- આકૃતિ 6.2ના ઊભા મેનૂમાં દર્શાવ્યા પ્રમાણે **Places → Search for Files** આદેશ આપો.
- જ્યારે તમે **Search for Files** (જુઓ આકૃતિ 6.2માં માઉસ તીરની નિશાની) ઉપર ક્લિક કરો છો ત્યારે ફાઈલની માહિતી પૂછતું એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થશે. આ માહિતીમાં ફાઈલનું નામ, જે ફોલ્ડરમાં ફાઈલ શોધવાની હોય તેનું નામ અને અપેક્ષિત ફાઈલની અંદરની કેટલીક માહિતીનો સમાવેશ થાય છે. આપણે છેલ્લે ફાઈલમાં સુધારા-વધારા કર્યાની તારીખ જેવી કેટલીક તારીખ વડે પણ ફાઈલને શોધી શકીએ છીએ. આકૃતિ 6.2માં ડાયલોગ બોક્સ નિર્દેશ કરતું તીર સાથેનું એક બોક્સ જુઓ.
- **Search for Files** ડાયલોગ બોક્સના **Name contains:** માં ફાઈલનું પૂર્ણ નામ અથવા નામનો થોડો ભાગ દાખલ કરો.
- હવે Find બટન ઉપર ક્લિક કરો.

આકૃતિ 6.2માં આ પગલાંઓ સચિત્ર દર્શાવ્યા છે. શોધવાની આ પ્રક્રિયાના પરિણામે ફાઈલની એક યાદી તે પ્રદર્શિત કરશે. તમે ફાઈલ ખોલવા માટે જરૂરી ફાઈલ ઉપર માઉસનો ઉપયોગ કરીને ડબલ ક્લિક કરો.



આકૃતિ 6.2 : ફાઈલ શોધવા માટેની શોધપ્રક્રિયાનું પરિણામ

સ્પ્રેડશીટનો સંગ્રહ કરવો અને બંધ કરવી (Saving and closing spreadsheet)

અગાઉના વિભાગમાં તમે કોઈ નવો દસ્તાવેજ અથવા હયાત દસ્તાવેજને જોવા માટે કે તેમાં સુધારા-વધારા માટે સ્પ્રેડશીટ દસ્તાવેજને તમે અત્યાર સુધીમાં ખોલેલો છે. ખુલ્લા દસ્તાવેજમાં એક વખત કામ પૂર્ણ થઈ જાય એટલે તેમાં કરેલાં પરિવર્તનોનો સંગ્રહ કરવાની તમને જરૂર પડે છે. કોઈ પણ ડેટા ગુમાવવાને અટકાવવા માટે કરેલાં પરિવર્તનોનો વારંવાર સંગ્રહ કરવો એ સલાહભર્યું છે.

આપણે પ્રકરણ 5માં વર્કશીટનો સંગ્રહ કઈ રીતે કરવો તે બાબત ચર્ચા કરી હતી. વર્કશીટનો સંગ્રહ કરવા માટે આદેશ આપો : **File → Save**. જો તમે દસ્તાવેજને અન્ય નવા નામથી કે અલગ પ્રકારની ફાઈલ તરીકે સંગ્રહ કરવા ઇચ્છતા હોય તો તમે **Save as** વિકલ્પને પણ પસંદ કરી શકો છો. અહીં અગાઉ જણાવ્યા પ્રમાણે, જો તમે સ્પ્રેડશીટનો પહેલી વખત સંગ્રહ કરો છો તો ફાઈલનું નામ અને તેને સંગ્રહ કરવાનો પાથ તમને પૂછશે.

જે ફાઈલ ખુલેલી છે (વર્તમાન) તેને બંધ કરવા માટે **File → Close** આદેશ આપો. જો તમે છેલ્લાં ફેરફારોનો સંગ્રહ ન કર્યો હોય તો કેલ્સી તમને ફાઈલ બંધ કરતા પહેલાં ફાઈલનો સંગ્રહ કરવાની એક તક આપશે.

વર્કશીટનો વિવિધ બંધારણ(ફોર્મેટ)માં સંગ્રહ (Saving worksheet in different format)

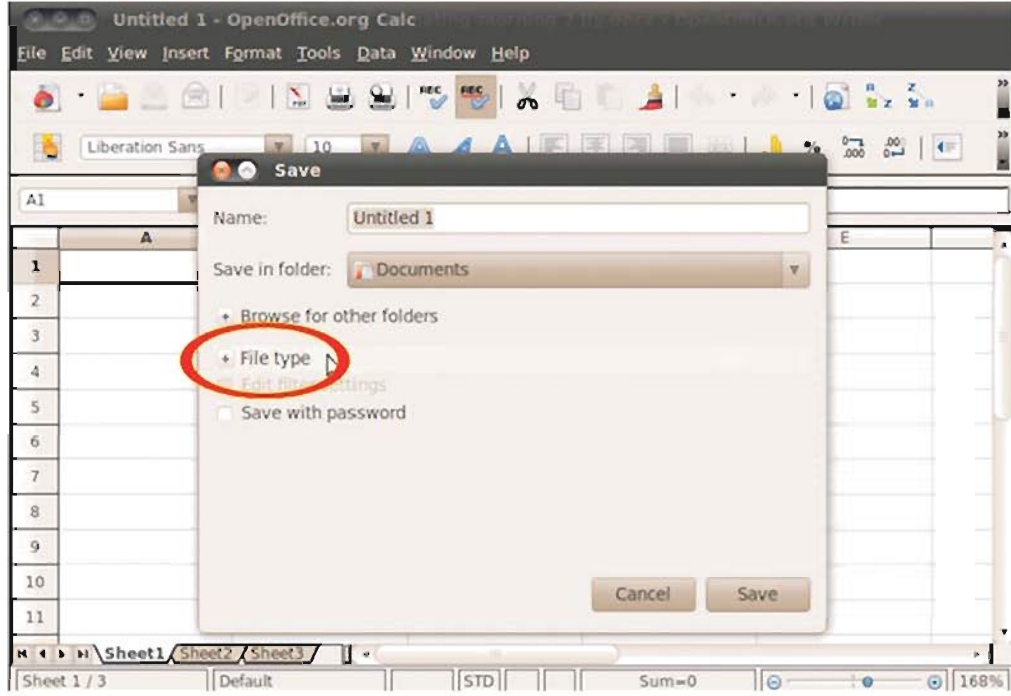
કેલ્સી પૂર્વનિર્ધારિત રીતે, સ્પ્રેડશીટને .ods અનુલંબન સાથે ફાઈલ સ્વરૂપે સંગ્રહ કરે છે. અન્ય કેટલાંક ફાઈલસ્વરૂપ (બંધારણ)ની યાદી કોષ્ટક 6.1માં આપી છે.

ફોર્મેટ (બંધારણ)	અનુલંબન	વર્ણન
ODF સ્પ્રેડશીટ	ods	OpenOffice.org calc ફોર્મેટ
ODF સ્પ્રેડશીટ ટેમ્પલેટ	ots	કેલ્સી સ્પ્રેડશીટ ટેમ્પલેટ ફોર્મેટ
dBASE	dbf	ડેટા બેઝ ફાઈલ ફોર્મેટ
શાબ્દિક CSV	csv	અલ્પવિરામથી કિંમતો અલગ પાડેલી શાબ્દિક ફાઈલ; ખાસ કરીને આવી ફાઈલ વિવિધ પ્રોગ્રામમાં માહિતી વિનિમય માટે વપરાય છે.
HTML દસ્તાવેજ (OpenOffice.org Calc)	html	વેબ પેઈજ ફોર્મેટ
સુવાહ્ય (પોર્ટેબલ) દસ્તાવેજ બંધારણ	pdf	સૌથી વધારે વપરાતું ફોર્મેટ; આ એક સુવાહ્ય દસ્તાવેજનું વિશ્વવ્યાપી એડોબ(Adobe)નું ફોર્મેટ છે.
માઈક્રોસોફ્ટ એક્સેલ 2007 XML	xlsx	એમએસઓફિસ 2007/2010ની સ્પ્રેડશીટ
માઈક્રોસોફ્ટ એક્સેલ 2003	xls	એમએસઓફિસ 2003ની સ્પ્રેડશીટ

કોષ્ટક 6.1 : વિવિધ ફાઈલ ફોર્મેટ કે જેમાં સ્પ્રેડશીટનો સંગ્રહ કરી શકાય

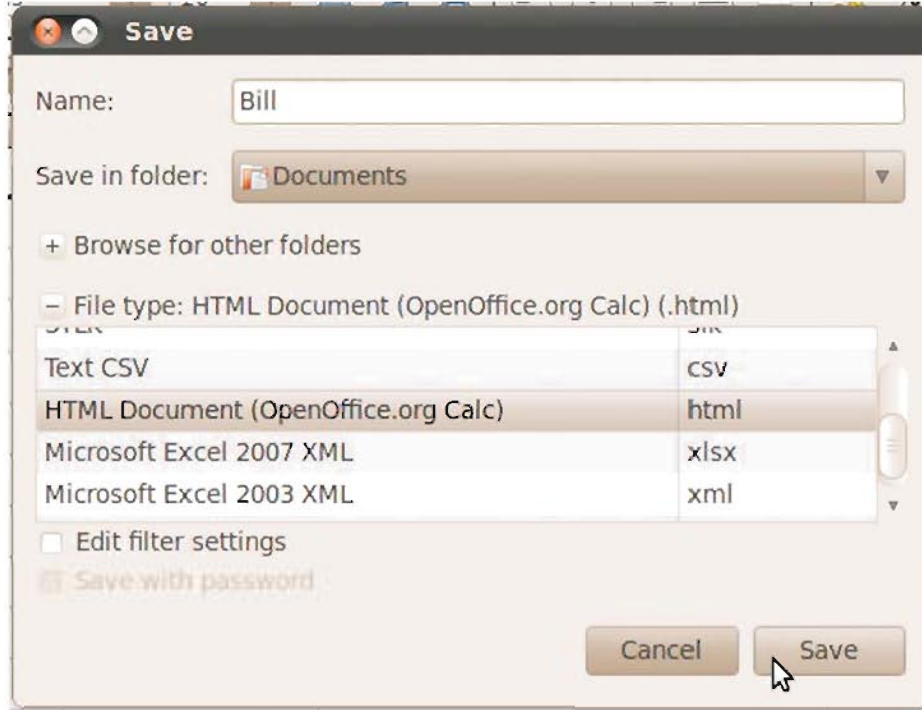
સ્પ્રેડશીટનો અન્ય ફાઈલ ફોર્મેટમાં સંગ્રહ કરવા માટે :

- સૌપ્રથમ File અને પછી Save (કે જે દસ્તાવેજનો અગાઉ સંગ્રહ કર્યો નથી) અથવા Save As વિકલ્પ (ખુલ્લા અને અગાઉ સંગ્રહ કરેલાં દસ્તાવેજ માટે) પસંદ કરો.
- આકૃતિ 6.3માં દર્શાવ્યા પ્રમાણે file type મેનૂ (નિર્દેશિત) ખોલો.



આકૃતિ 6.3 : file type મેનૂ ખોલવું

- આકૃતિ 6.4માં દર્શાવ્યા પ્રમાણે જરૂરી ફાઈલ ફોર્મેટની પસંદગી કરો.



આકૃતિ 6.4 : ફાઈલના પ્રકારની પસંદગી કરવી

- **Save** બટન ક્લિક કરી ફાઈલનો સંગ્રહ કરો.

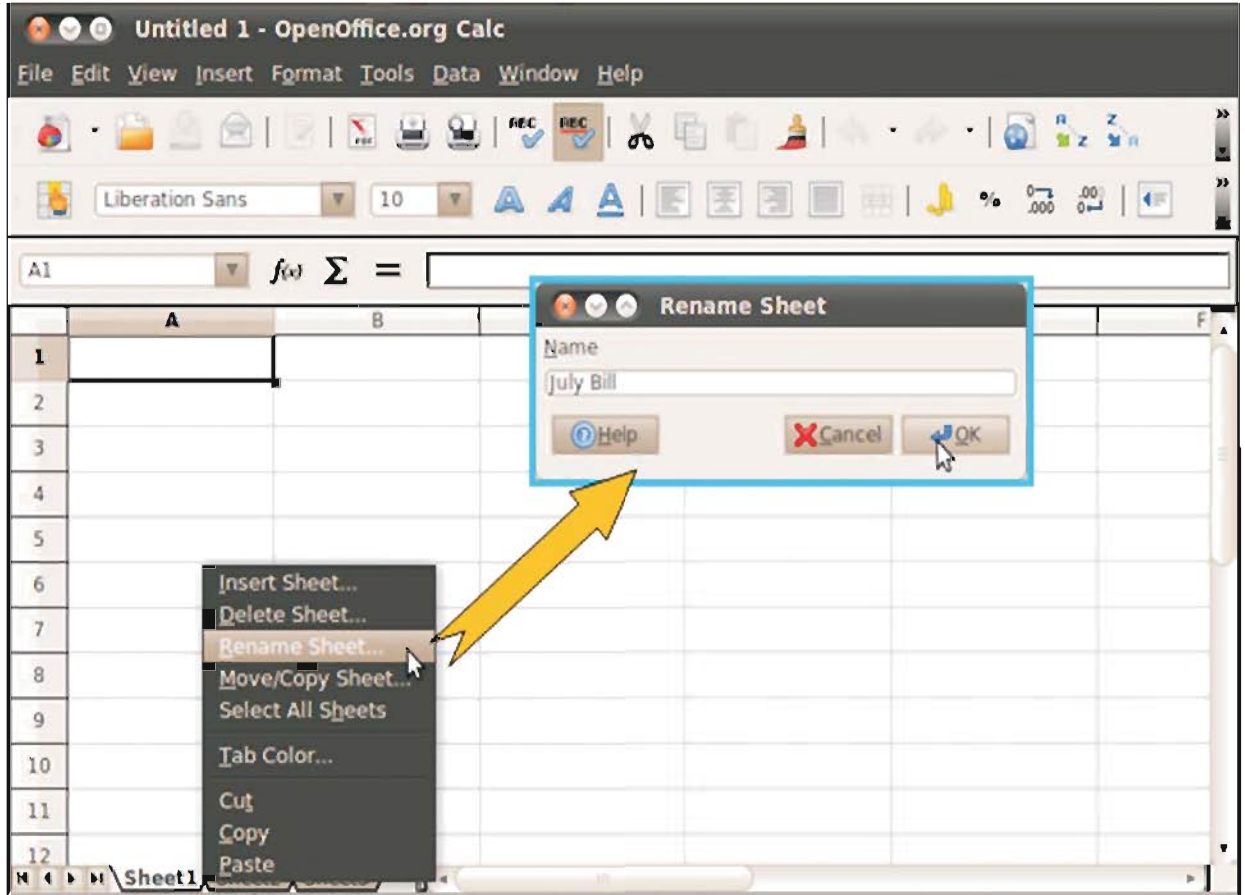
વર્કશીટનું નામ અને રંગ બદલવા (Renaming and re-colouring worksheet)

આપણે અગાઉના પ્રકરણમાં સ્ટેટસ બાર ઉપર રાઈટ ક્લિક કરી વર્કશીટ પસંદ કરીને વર્કશીટનું નામ બદલ્યું હતું. જો સ્પ્રેડશીટ દસ્તાવેજમાં અનેક વર્કશીટ તમારી પાસે હોય તો તે દસ્તાવેજના સરળ સંચાલન માટે તેને યોગ્ય નામ (અને રંગ) આપવા વધુ સારા છે. વર્કશીટનું નામ બદલવા માટે નીચે પ્રમાણે ક્રિયાઓ કરો :

- ખુલ્લી વર્કશીટમાંથી કોઈ પણ એક સેલની પસંદગી કરો.
- હવે આદેશ આપો : **Format → Sheet → Rename**
- પ્રદર્શિત **Rename Sheet** ડાયલોગ બોક્સમાં નવું નામ દાખલ કરો.
- **OK** બટન દબાવીને પરિવર્તનોનો સંગ્રહ કરો.

આના અન્ય વિકલ્પરૂપે શીટ ટેબ ઉપર દર્શાવેલાં વર્કશીટનાં નામ ઉપર રાઈટ ક્લિક કરીને નીચે દર્શાવ્યા પ્રમાણે Rename sheet વિકલ્પ પસંદ કરો. આ વિકલ્પ વિશે તમે અગાઉ શીખ્યા છો. આ ક્રિયા નીચે જણાવેલ છે :

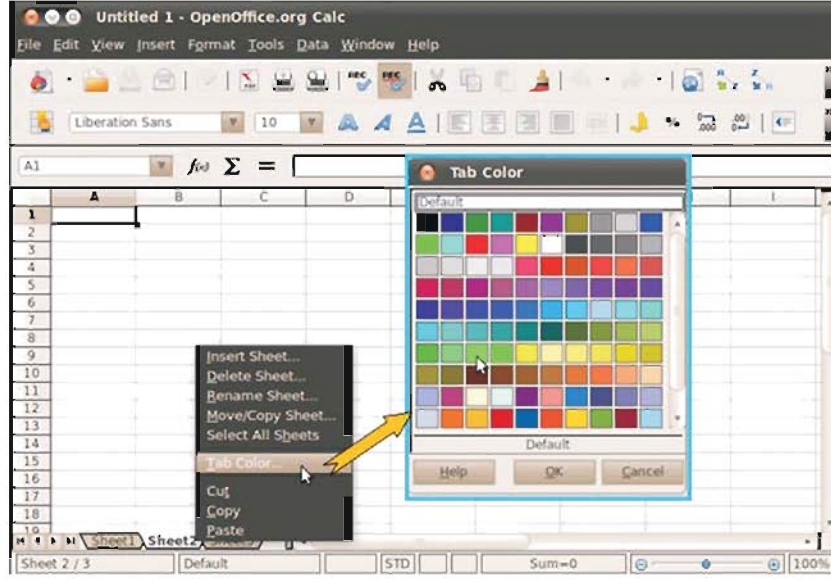
- તમારે જે વર્કશીટનું નામ બદલવું હોય તે શીટ પસંદ કરો. પસંદ કરેલી શીટ હાઈલાઈટ થયેલી દેખાશે (સફેદ પશ્ચાદ્ભૂમિ સાથે)
- તે શીટ ઉપર રાઈટ ક્લિક કરો. આકૃતિ 6.5માં દર્શાવ્યા પ્રમાણે એક ઊભું મેનૂ પ્રદર્શિત થશે તેમાંથી Rename sheet વિકલ્પ પસંદ કરો, આથી આકૃતિ 6.5ની મધ્યમાં લંબચોરસ આકારનું એક નવું ડાયલોગ બોક્સ દેખાય છે.
- હવે શીટને નવું નામ આપો.



આકૃતિ 6.5 : શીટ ટેબ વડે વર્કશીટનું નામ બદલવું

વર્કશીટ ટેબનો રંગ બદલવો (Changing colour of the worksheet tab)

વર્કશીટનાં ટેબનો રંગ બદલવા માટે આકૃતિ 6.6માં દર્શાવ્યા પ્રમાણે આપણે ફક્ત વર્કશીટના નામ ઉપર રાઈટ ક્લિક કરીને સ્ક્રીન ઉપર પ્રદર્શિત ઊભા મેનૂમાંથી **Tab color** આદેશ પસંદ કરવો પડે છે. આપણી સામે રજૂ કરેલા વિવિધ રંગોમાંથી આપણે ટેબ માટે યોગ્ય રંગ પસંદ કરવો પડે છે.



આકૃતિ 6.6 : વર્કશીટ ટેબનો રંગ બદલવો

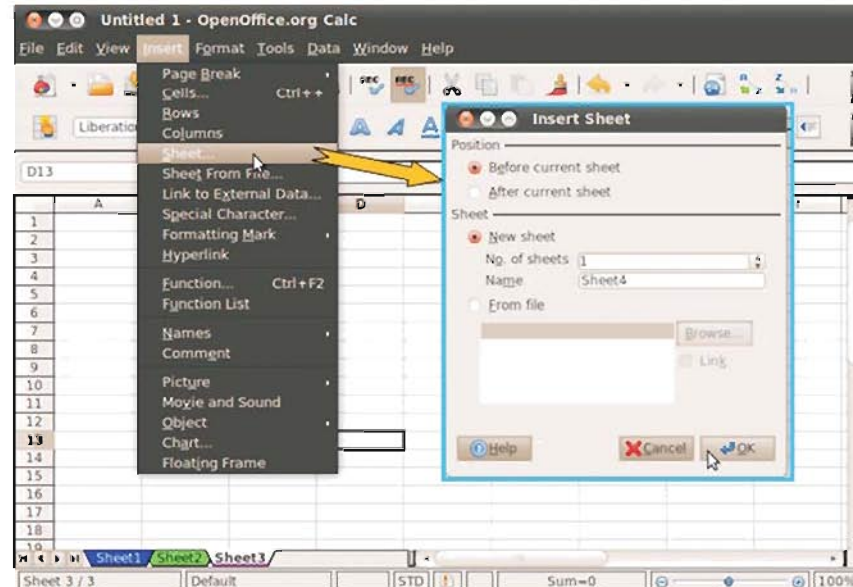
વર્કશીટ ઉમેરવી અને દૂર કરવી (Inserting and deleting worksheets)

તમે જાણતા હશો કે ઘણીવાર અમુક વિનિયોગ સંબંધિત બધો ડેટા દાખલ કરવા માટે કેલ્સી દ્વારા પૂરી પાડવામાં આવતી ત્રણ વર્કશીટ પૂરતી નથી. આવા સંજોગોમાં, આપણે વધારાની વર્કશીટ ઉમેરવી પડે છે. દા.ત. કોઈ પ્રવૃત્તિ, જેમ કે રમતગમતમાં વિદ્યાર્થીઓનાં પાંચ જૂથ છે. દરેક જૂથ માટે એક હાજરીપત્રક બનાવવું છે. આ ઉદાહરણમાં પાંચ વર્કશીટ હોય તે વધુ ઇચ્છનીય છે; દરેક વર્કશીટ જૂથદીઠ અલગ અલગ હાજરીપત્રક હોય. ફોર્મેટ કરેલી સ્પ્રેડશીટમાં રો અને કોલમ ઉમેરી શકાય છે, પહોળાઈ બદલી શકાય છે અને સેલને ભેગા કરી શકાય છે.

નવી વર્કશીટ ઉમેરવા માટે નીચે પ્રમાણે ક્રિયાઓ કરો :

- એ વર્કશીટનો કોઈ પણ સેલ પસંદ કરો કે જે વર્કશીટની આગળ કે પાછળ નવી વર્કશીટ ઉમેરવાની હોય.
- હવે આદેશ આપો: **Insert → Sheet**. આથી **Insert Sheet** ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થશે.
- આ ઇન્સર્ટ શીટ (Insert Sheet) ડાયલોગ બોક્સમાં વધારાના વિકલ્પો જણાવો.
- અંતમાં, આ ક્રિયાને મંજૂર કરાવવા માટે **OK** બટન ઉપર ક્લિક કરો.

વધારાની નવી વર્કશીટ ઉમેરવા માટેની ઉપર જણાવેલી ક્રિયાઓ આકૃતિ 6.7માં સચિત્ર દર્શાવેલી છે.



આકૃતિ 6.7 : દસ્તાવેજમાં નવી વર્કશીટ ઉમેરવી

તમે કદાચ વર્કશીટ ટેબ મારફતે નવી વર્કશીટ ઉમેરવાની શક્યતા બાબતે પણ વિચાર્યું હશે. શીટ ટેબનો ઉપયોગ કરીને વર્કશીટ ઉમેરવા માટે નીચે જણાવેલાં પગલાંને અનુસરો :

- વર્કશીટ પસંદ કરીને તેના ઉપર રાઈટ ક્લિક કરો.
- એક ઊભું મેનૂ ખૂલશે.
- હવે **Insert Sheet** પસંદ કરો.
- તમે Insert Sheet ડાયલોગ બોક્સ જોઈ શકશો.
- જરૂરી વિકલ્પો આપીને OK બટન ઉપર ક્લિક કરો.

વર્કશીટ દૂર કરવી (Deleting worksheet)

કેલ્સીમાં તમે એક અથવા વધારે વર્કશીટ દૂર કરી શકો છો. કોઈ એક વર્કશીટ દૂર કરવા માટે તમે જે દૂર કરવા ઇચ્છતા હોય તે વર્કશીટના શીટ ટેબ ઉપર રાઈટ ક્લિક કરો. તે તમને શક્ય ક્રિયાઓની યાદી જણાવશે. આ વિકલ્પોમાંથી **Delete Sheet** પસંદ કરો. આના વિકલ્પરૂપે, તમે આ પ્રમાણે આદેશ આપો : **Edit → Sheet → Delete**

રો અને કોલમ કક્ષાની ક્રિયાઓ (Row and Column level operations)

અમુક સમયે આપણે વર્કશીટમાં કેટલીક કોલમ કે રો ઉમેરવાની જરૂર પડે છે. જો કે વર્કશીટ ઘણી રો અને કોલમ પૂરી પાડે છે પણ એ શક્ય છે કે આપણે દાખલ કરેલા ડેટાની વચ્ચે કેટલીક ખાલી રો કે કોલમ ઉમેરવા ઇચ્છીએ. આપણે આ વિભાગમાં રો અને કોલમ કક્ષાની અન્ય ક્રિયાઓ સાથે વર્કશીટમાં રો અને કોલમ કઈ રીતે ઉમેરી શકીએ તેની ચર્ચા કરીશું. વર્કશીટમાં રો (આડી હરોળ) ઉમેરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- જે રો ની ઉપર નવી રો ઉમેરવાની હોય તે રો માંથી કોઈ એક સેલ પસંદ કરો.
- હવે આદેશ આપો : **Insert → Rows**

વર્કશીટમાં કોલમ (ઊભી હરોળ) ઉમેરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- જે કોલમની ડાબી બાજુમાં નવી કોલમ ઉમેરવાની હોય તે કોલમમાંથી એક સેલ પસંદ કરો.
- હવે આદેશ આપો : **Insert → Column**

કોલમની પહોળાઈ અને રો ની ઊંચાઈ બદલવી (Width of a column and height of a row)

કેલ્સીની બધી જ રો (આડી હરોળ) અને કોલમ (ઊભી હરોળ) એક્સરખી ઊંચાઈ અને પહોળાઈ ધરાવે છે. જ્યારે તમે સેલમાં શબ્દિક માહિતી લખો છો ત્યારે ઘણી વખત બાજુના સેલની માહિતીને કારણે સેલની માહિતી અદૃશ્ય થઈ જાય છે. અમુક સમયે કેટલાંક સૂત્રોનું પરિણામ સેલમાં સમાતું પણ નથી. કોલમની પહોળાઈ અથવા રોની ઊંચાઈ બદલવા માટે નીચે પ્રમાણે કાર્ય કરો :

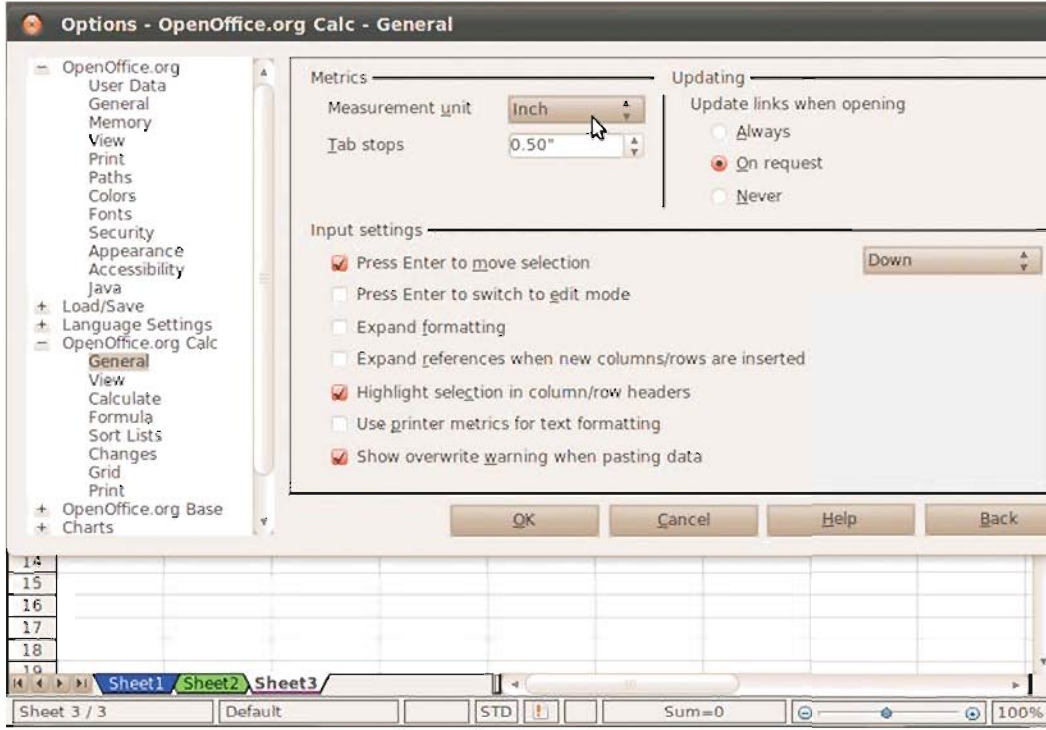
- કોલમ(રો)નાં શીર્ષકોને જુદી પાડતી રેખા ઉપર ક્લિક કરો.
- માઉસનું ડાબું બટન છોડી દીધા વિના તમે જરૂરી દિશામાં ડ્રેગ કરો (ખેંચો).

કોલમ અથવા રોને કોઈ ચોક્કસ માપ આપવા માટે નીચે પ્રમાણે કાર્ય કરો :

- કોલમ(રો)માંથી કોઈ પણ એક સેલ પસંદ કરો.
- હવે આદેશ આપો : **Format → Column → Width** અથવા **Format → Row → Height**
- સ્ક્રીન ઉપર જે ડાયલોગ બોક્સ પ્રદર્શિત થાય તેમાં કોલમની પહોળાઈ અથવા રો ની ઊંચાઈ ઈંચમાં જણાવો.
- હવે **OK** બટન દબાવો.

જો તમને ઈંચમાં માપ આપવાનું અનુકૂળ ન હોય તો નીચે પ્રમાણે આદેશ આપો :

Tools → Options → General આકૃતિ 6.8માં દર્શાવ્યા પ્રમાણે તમારી જરૂરિયાત મુજબ તે વર્ગ(General વર્ગ)માં પૂરા પાડવામાં આવેલાં વિકલ્પો બદલો.



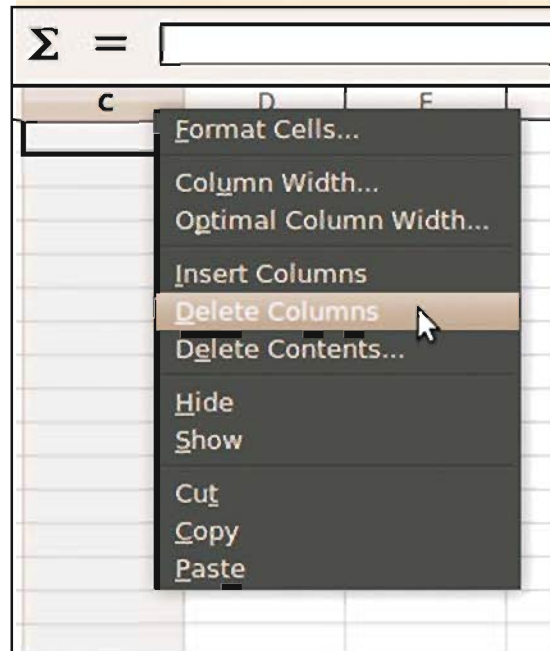
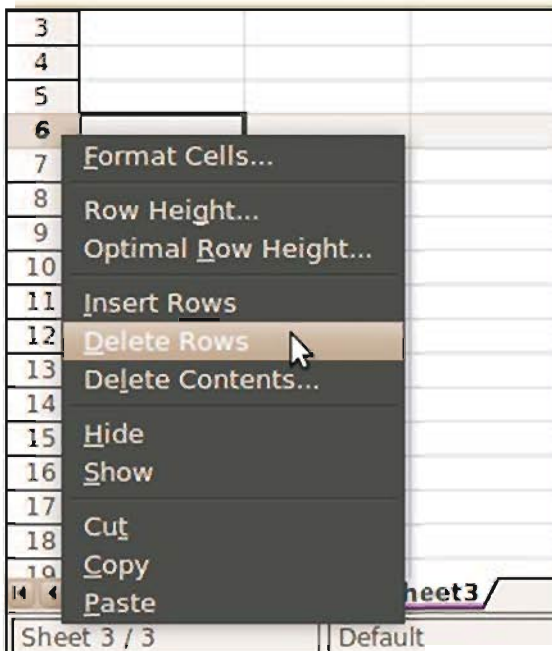
આકૃતિ 6.8 : કેલ્સીમાં General ના વિકલ્પો

રો અને કોલમ દૂર કરવી (Deleting rows and columns)

રો દૂર કરવા માટે નીચે જણાવેલાં પગલાં ભરો :

- તમે જે રો દૂર કરવા ઇચ્છતા હોય તે રો ઉપર તમે જાઓ.
- તે રો ઉપર રાઈટ ક્લિક કરો.
- હવે **Delete rows** પસંદ કરો.

આ જ રીતે તમે કોલમ પણ દૂર કરી શકો છો. આકૃતિ 6.9માં રો અને કોલમને દૂર કરવાની પ્રક્રિયાનું પ્રદર્શન કરેલું છે.



આકૃતિ 6.9 : રો અને કોલમને દૂર કરવી

અન્ય વિકલ્પરૂપે, રો કે કોલમ દૂર કરવા માટે નીચે જણાવેલાં પગલાંઓને અનુસરો :

- સૌપ્રથમ કોલમ અથવા રો પસંદ કરો.
- **Edit → Delete Cells** પસંદ કરો.

આ ક્રિયા આકૃતિ 6.10માં સચિત્ર દર્શાવેલી છે. આ વિકલ્પ વડે આપણી જરૂરિયાત પ્રમાણે સેલને ખસેડીને આખી કોલમ અથવા રો ને દૂર કરવી શક્ય છે.

રો અને કોલમને પસંદ કરવી (Selecting rows and columns)

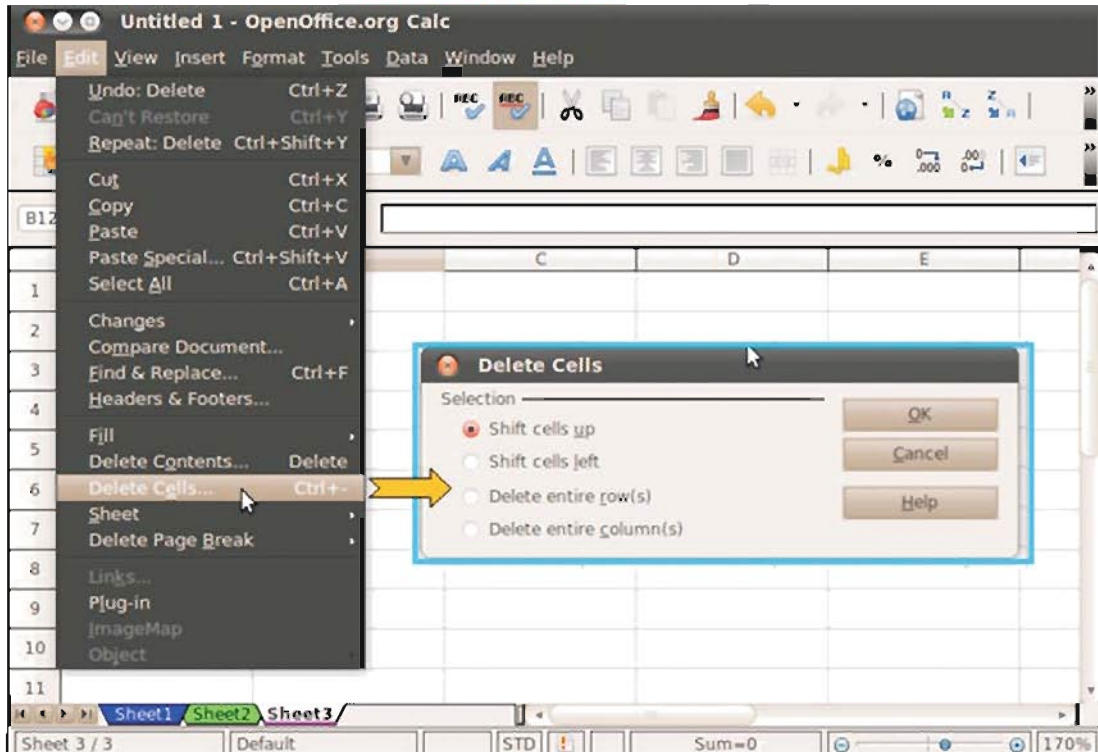
આખી ઊભી હરોળને પસંદ કરવા માટે કોલમ હેડર (column header) ઉપર ક્લિક કરો, તમે જોઈ શકશો કે આખી ઊભી હરોળ હાઈલાઈટ થયેલી છે. આ જ પ્રમાણે, આખી આડી હરોળને પસંદ કરવા માટે રો હેડર (row header) ઉપર ક્લિક કરો અને આથી આખી આડી હરોળ હાઈલાઈટ થયેલી જોવા મળશે.

વિભાગ સ્થગિત બનાવવો (Freezing a pane)

જ્યારે એક જ વર્કશીટમાં સંખ્યાબંધ રો અને કોલમનો સમાવેશ કરવામાં આવે છે ત્યારે કોલમ હેડર અથવા રો હેડર સ્ક્રીન ઉપર દેખાતું રહેતું નથી કે જેથી ડેટાનું સંચાલન મુશ્કેલ બની જાય છે. કોલમ હેડર અને રો હેડર હંમેશાં દૃશ્યમાન રહે તે માટે નીચે મુજબ કાર્ય કરો :

- બધા કોલમ હેડર ધરાવતી આડી હરોળની નીચેની રોમાંથી કોઈ એક સેલ પસંદ કરો. કોલમના કિસ્સામાં જે કોલમ હંમેશાં દૃશ્યમાન રાખવી હોય તેની જમણી બાજુની કોલમ પસંદ કરો.
- હવે પસંદ કરો : **Window → Freeze**

સેલને ચલિત (unfreeze) બનાવવા માટે **Window → Freeze** આદેશ ફરી આપીને Freeze વિકલ્પ નાપસંદ કરો.



આકૃતિ 6.10 : સેલને દૂર કરવા

સેલ કક્ષાની ક્રિયાઓ (Cell Level Operations)

આ વિભાગ તમને સેલમાં કઈ રીતે માહિતી લખી શકાય તેની સમજ આપશે. આ માહિતી શાબ્દિક લખાણ, સંખ્યા, વિધેયો, સૂત્રો કે અન્ય સેલનો સંદર્ભ હોઈ શકે. આ ઉપરાંત તમે એક સેલ કે અનેક સેલની માહિતીની નકલ કેવી રીતે કરવી અને કેવી રીતે ખસેડવી તે બાબત પણ શીખશો.

સેલ પસંદ કરવો (Selecting cells)

ઉપર જણાવ્યા પ્રમાણે સેલમાં શાબ્દિક લખાણ, સંખ્યા, વિધેયો, સૂત્રો કે અન્ય સેલનો સંદર્ભ દાખલ કરી શકાય. પણ સેલમાં કંઈ પણ દાખલ કરતાં પહેલાં આપણે સેલને પસંદ કરવો પડે. નહિતર ઈચ્છિત સેલમાં લખાણ ઉમેરાશે નહીં. સેલને પસંદ કરવા માટે ફક્ત તમારે તેના ઉપર લેફ્ટ ક્લિક કરવું પડશે. તે પછી સેલમાં તમે લખાણ કરી શકશો.

સેલનો વિસ્તાર પસંદ કરવો (Selecting a range of Cells)

કોઈ સેલ વિસ્તાર પસંદ કરવા માટે તે વિસ્તારના પહેલા સેલ ઉપર જઈ લેફ્ટ ક્લિક કરો. માઉસને લેફ્ટ ક્લિક કરીને પહેલાં સેલ પસંદ કર્યા પછી માઉસનું બટન છોડી દીધા વિના માઉસ પોઈન્ટરને ઈચ્છિત વિસ્તારના છેલ્લા સેલ સુધીના બધા સેલનો સમાવેશ થાય ત્યાં સુધી ડ્રેગ કરો. એક સમયે વિસ્તારના બધા સેલ હાઈલાઈટ થઈ જાય પછી માઉસ બટનને છોડી દો.

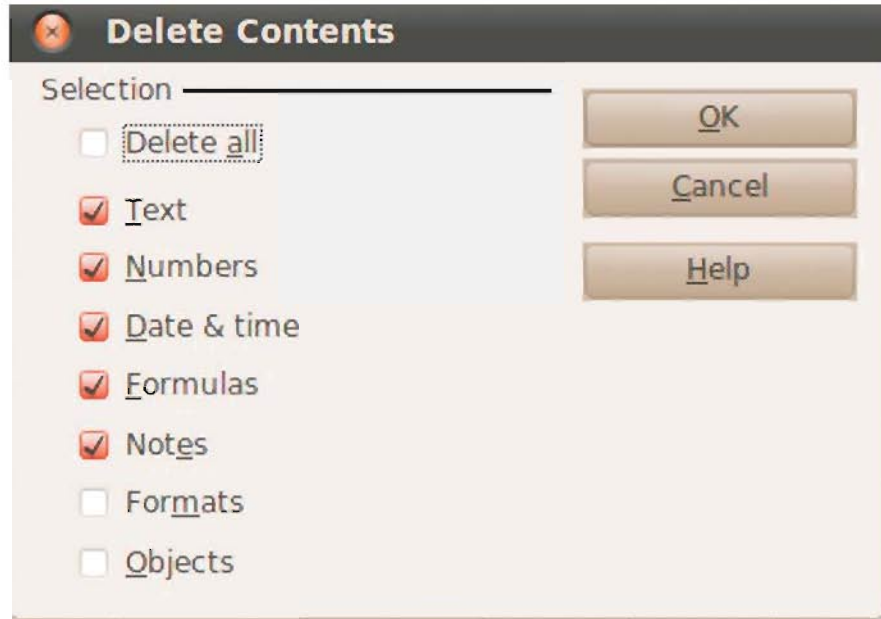
ઉપર જણાવેલી પ્રક્રિયા ફક્ત એકબીજાની બાજુમાં રહેલા સેલને લાગુ પડે છે. જો સેલ ભૌતિક રીતે એકબીજાને અડીને ન હોય તો તમારે **CTRL** કી દબાવીને રાખવી પડે છે. જો તમારે આખી વર્કશીટ (વર્કશીટના બધા સેલ) પસંદ કરવી હોય તો તમારે રો હેડર અને કોલમ હેડરની શરૂઆતનું બટન જ ફક્ત ક્લિક કરવું પડે છે. રો અને કોલમના નિર્દેશો આપીને પણ તમે વિસ્તાર પસંદ કરી શકો છો જેમ કે એડ્રેસ બોક્સમાં A1:B12 જણાવીને.

કોઈ એક સેલ અથવા સેલના વિસ્તારની પસંદગી રદ કરવા માટે તમારે ફક્ત માઉસ વડે કોઈ પણ સેલ ઉપર લેફ્ટ ક્લિક કરવું પડે છે.

સેલની માહિતી રદ કરવી (To delete the content of a cell)

કોઈ એક સેલ કે સેલના વિસ્તારમાં રહેલી માહિતી દૂર કરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- એક સેલ અથવા સેલનો વિસ્તાર પસંદ કરો.
- કી બોર્ડ ઉપરની ડિલીટ કી (**Delete**) દબાવો આથી સ્ક્રીન ઉપર એક ડાયલોગ બોક્સ પ્રદર્શિત થશે.
- ક્લિક કરીને તમારી અલગ અલગ પસંદ જણાવો. જો તમે શાબ્દિક લખાણ રદ કરવા ઈચ્છતા હોય તો **Text** ઉપર ક્લિક કરો. તમે એક કરતાં વધારે પસંદગી આપી શકો છો.
- આકૃતિ 6.11માં દર્શાવ્યા પ્રમાણે **Delete Contents** ડાયલોગ બોક્સનાં **OK** બટન ઉપર ક્લિક કરીને તમારી પસંદગીની પુષ્ટિ કરો.



આકૃતિ 6.11 : સેલની માહિતી રદ કરવાના વિવિધ વિકલ્પો

સ્ક્રીન ઉપર પ્રદર્શિત ડાયલોગ બોક્સ રો અને કોલમ રદ કરવાના વિવિધ વિકલ્પો રજૂ કરે છે. કોષ્ટક 6.2માં **Delete Contents** ડાયલોગ બોક્સમાં ઉપલબ્ધ વિકલ્પોની યાદી રજૂ કરેલી છે.

વિકલ્પ	વર્ણન
Delete all	સેલ / સેલના વિસ્તારની સંપૂર્ણ માહિતી રદ કરે છે.
Text	પસંદ કરેલા સેલ / સેલ વિસ્તારનું શાબ્દિક લખાણ રદ કરે છે.
Numbers	પસંદ કરેલા સેલ / સેલ વિસ્તારમાંથી સંખ્યા રદ કરે છે.
Date & time	પસંદ કરેલા સેલ / સેલ વિસ્તારમાંથી ફક્ત તારીખ અને સમય રદ કરે છે.
Formulas	પસંદ કરેલા સેલ / સેલ વિસ્તારમાંથી સૂત્રો અને તેનાં પરિણામો રદ કરે છે.
Notes	જો કોઈ સેલ માટે નોંધ ઉમેરી હોય તો રદ કરે છે.
Formats	સેલ ફોર્મેટિંગ (માળખું) રદ કરે છે પણ સેલનો ડેટા રાખે છે.
Objects	સેલનાં વધારાનાં તત્વો રદ કરે છે - જેમ કે ઇમેજ.

કોષ્ટક 6.2 : Delete Contents ડાયલોગ બોક્સના ઉપલબ્ધ વિકલ્પો

સેલ માહિતીમાં સુધારા-વધારા કરવા (Editing the cell content)

સેલમાં રહેલી માહિતીમાં ફેરફાર કરવા (એડિટ - edit) માટે ફક્ત સેલને પસંદ કરો અને માઉસ વડે ડબલ ક્લિક કરીને માહિતીમાં સુધારા કરો. જ્યારે સુધારા કરવાનું કાર્ય પૂર્ણ થાય તે પછી એન્ટર કી દબાવો.

સેલની નકલ કરવી અને તેને ખસેડવા (Copying and moving cells)

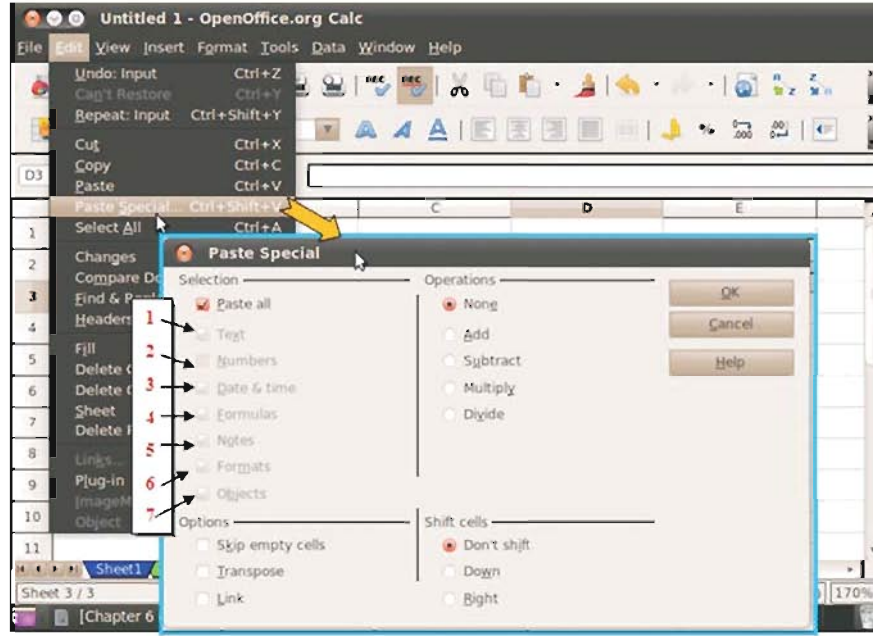
સ્પ્રેડશીટના સેલમાં અનેક પ્રકારનો ડેટા હોય છે, જેમ કે સંખ્યા, શાબ્દિક લખાણ, સૂત્રો, ફોર્મેટ (માળખું) અને અન્ય સેલનો નિર્દેશ. આ રીતે સેલની માહિતી જટિલ છે, કારણ કે તે અનેક પ્રકારની કિંમતો ધરાવે છે. કોઈ એક સેલ, વર્કશીટ કે સેલ વિસ્તારની નકલ કરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- તમે જેની નકલ કરવા ઇચ્છતા હોય તે સેલ / સેલનો ભાગ / સેલનો વિસ્તાર / વર્કશીટને પસંદ કરો.
- આદેશ આપો : **Edit → Copy** આના વિકલ્પો તમે સેલ પસંદ કરીને તેના ઉપર રાઈટ ક્લિક પણ કરી શકો છો અને તે પછી મેનૂમાંથી **Copy** વિકલ્પ આપો. નકલ થયેલો ડેટા ટપકાં ટપકાંથી બનેલા લંબચોરસમાં હાઈલાઈટ થયેલો દેખાશે.
- જે જગ્યાએ ડેટાની નકલ કરવાની હોય તે જગ્યા નક્કી કરો અને **Paste** આદેશ આપો.

Paste આદેશનાં વધારાના વિકલ્પો માટે આકૃતિ 6.12માં દર્શાવ્યા પ્રમાણે **Edit → Paste Special** આદેશ આપો. Paste Special ડાયલોગ બોક્સ જે માહિતીની નકલ કરવાની છે તેના માટે અનેક પરિમાણો (parameters) ધરાવે છે. આ પસંદગીઓની યાદી કોષ્ટક 6.3માં આપેલી છે.

લેબલ	વર્ણન
1	ફક્ત શાબ્દિક લખાણની નકલ (paste) કરે છે.
2	ફક્ત સંખ્યાની નકલ (paste) કરે છે.
3	ફક્ત તારીખ અને સમયની નકલ (paste) કરે છે.
4	ફક્ત સૂત્રોની નકલ (paste) કરે છે.
5	ફક્ત નોંધની નકલ (paste) કરે છે.
6	કોઈ કિંમત અથવા સેલનું ફોર્મેટની નકલ (paste) કરે છે.
7	ફક્ત ઓબ્જેક્ટની નકલ (paste) કરે છે.

કોષ્ટક 6.3 : Paste Specialની પસંદગીઓ



આકૃતિ 6.12 : Paste Special ના વિકલ્પો

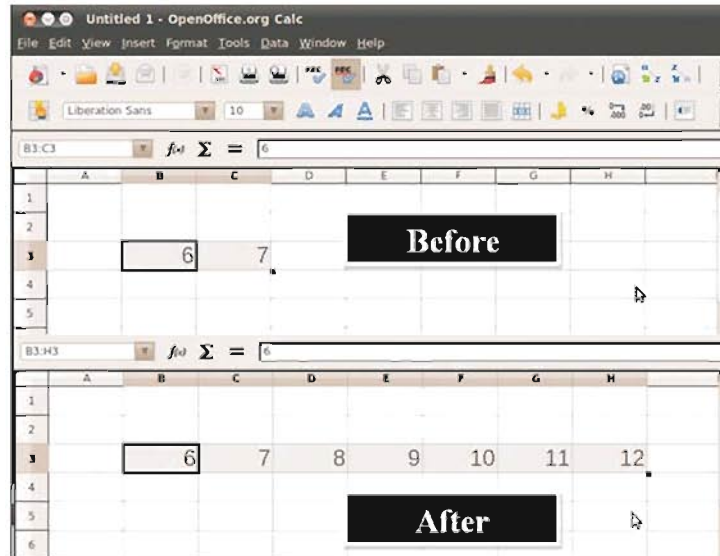
Paste Special ક્રિયા વડે તમે ગંતવ્ય સેલ ઉપર માહિતી કેવી રીતે નકલ કરવી તેનું નિયંત્રણ કરી શકો છો.

ઑટોફિલ ટૂલ (Autofill tool)

સેલમાં આપમેળે ડેટા ભરવા માટે ઑટોફિલ ટૂલ વપરાય છે. નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરો :

- પ્રથમ સેલમાં સંખ્યા દાખલ કરો.
- તે પછીના સેલમાં (રો કે કોલમ) ત્યારપછીની સંખ્યા દાખલ કરો.
- બંને સેલ પસંદ કરો.
- સેલનાં હેન્ડલ (handle) ઉપર લેફ્ટ ક્લિક કરો.
- માઉસ બટન દબાવેલું રાખીને ઇચ્છિત સેલ વિસ્તાર સુધી ડ્રેગ (ખેંચો) કરો.
- હવે માઉસ બટન છોડી દો.

આકૃતિ 6.13માં દર્શાવ્યા પ્રમાણેનો સ્ક્રીન તમે જોઈ શકશો.



આકૃતિ 6.13 : આપમેળે ભરાયેલી સંખ્યા

ઉપર જણાવેલી આપમેળે ભરાયેલી સંખ્યા સમાંતર શ્રેણી તરીકે પણ ઓળખાય છે. હવે તમે નીચે જણાવ્યા પ્રમાણે અલગ રીતે સમાંતર શ્રેણીનો પ્રયત્ન કરો :

તમે સેલમાં એકી સંખ્યા (ધારો કે 1) લખો. તેની બાજુના સેલમાં ક્રમિક એકી સંખ્યા (આપણા ઉદાહરણમાં અહીં પછીની એકી સંખ્યા 3 છે) લખો. હવે બંને સેલને એકસાથે પસંદ કરી તે પછીના 10 સેલમાં ઓટોફિલ ટૂલ વડે ભરવા પ્રયત્ન કરો. હવે તમે આપમેળે ભરાયેલી સંખ્યાનું નિરીક્ષણ કરો. તે તમામ એકી સંખ્યા છે ?

હવે અન્ય સંખ્યાઓની શ્રેણી લઈને પ્રયોગ કરી જુઓ. જેમ કે :

- 10, 20, 30,
- 25, 50, 75,...
- 2, 4, 5,....

આ ટૂલ વાપરીને તમે સંખ્યાના કોષ્ટક ($1*1=1$, $2*1=2$, ...) બનાવી શકો.

જો તમે ફક્ત એક જ સેલની સંખ્યા સાથે ઓટોફિલ ટૂલનો ઉપયોગ કરશો તો 1ના વધારા સાથેની સંખ્યાની યાદી મેળવશો. ફક્ત વિચારો કે સેલમાં સંખ્યાની જગ્યાએ કોઈ સેલ એડ્રેસ કે સેલ નિર્દેશ હોય તો શું થાય ? સેલમાં કેટલાક સેલ એડ્રેસ લખો અને બાજુના કેટલાંક સેલમાં ઓટોફિલ કરવાનો પ્રયત્ન કરી જુઓ.

સાપેક્ષ સ્થાનાંક અને નિરપેક્ષ સ્થાનાંક (Relative and absolute address)

અગાઉના ઉદાહરણમાં આપણે એક સેલના સૂત્રની નકલ બીજા અનેક સેલમાં કરી હતી. જો કોઈ સેલમાં સૂત્ર લખેલું હોય અને તે સેલની નકલ આપણે બીજા સેલમાં કરીએ તો સૂત્રમાં સામેલ સેલનાં સ્થાનાંક(એડ્રેસ-address)ને કેલ્સી આપોઆપ બદલી નાખે છે. (પ્રકરણ 5માં ખરીદીના બિલનાં ઉદાહરણમાં આપણે તે જોયું હતું.)

સેલ A1માં સેલ એડ્રેસ = C1 છે. આનો અર્થ એ થાય કે સેલ A1 સેલ C1ની માહિતીનો ઉલ્લેખ કરે છે; એટલે કે સેલ A1થી ત્રીજા સેલનો. હવે જો તમે સેલ A1ની નકલ સેલ B1માં કરો તો સેલ B1 તેનાથી ત્રીજા સેલ એટલે કે D1નો નિર્દેશ કરશે. આવા પ્રકારની સ્થાનાંક(એડ્રેસ)ની પ્રક્રિયાને સાપેક્ષ કહેવામાં આવે છે. જ્યારે સાપેક્ષ સ્થાનાંકની નકલ એક સેલમાંથી બીજા સેલમાં કરવામાં આવે ત્યારે તે આપોઆપ બદલાઈ જાય છે. આ જાતની રચના ટાળવા માટે અને સ્થાનાંકને સ્થાયી સ્વરૂપ આપવા માટે આપણે =\$A\$1 (રો અને / અથવા કોલમની પહેલાં ડોલરનું ચિહ્ન (\$) ઉમેરીને) જેવું સ્થાનાંક લખવું પડે. આ પ્રકારના સ્થાનાંકને નિરપેક્ષ સ્થાનાંક કહેવામાં આવે છે. જ્યારે સૂત્રની નકલ કરવામાં આવે કે ખસેડવામાં આવે ત્યારે નિરપેક્ષ સ્થાનાંકમાં કોઈ પરિવર્તન આવતું નથી. નિરપેક્ષ સ્થાનાંક હંમેશાં કોઈ ચોક્કસ સેલની કિંમતનો નિર્દેશ કરે છે. ઉદાહરણ તરીકે, કોઈ પદાવલિમાં સ્થાનાંક =\$A\$4 હંમેશાં સેલ A4ની કિંમત મેળવશે, જ્યારે પણ તેની નકલ કરવામાં આવે કે ખસેડવામાં આવે.

તમે કોઈ કોલમની આગળ \$ લખીને કોલમને અચલ અને રો ને ચલિત અથવા એથી વિપરીત પણ કરી શકો. ઉદાહરણ તરીકે, જો તમે કોઈ સ્થાનાંક =A\$4 તરીકે લખો અને તેને બીજી જગ્યાએ નકલ કરવાનો પ્રયત્ન કરો તો ફક્ત કોલમ બદલાશે પણ રો નહીં બદલાય. તેમાં હંમેશાં રો 4 રહેશે.

સૂત્ર સાથેના સેલમાં ઓટોફિલ કરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- સૂત્ર સાથેનો સેલ પસંદ કરો.
- સેલના હેન્ડલ ઉપર લેફ્ટ ક્લિક કરો.
- માઉસ બટનને દબાવી રાખીને ઇચ્છિત સેલ વિસ્તાર સુધી ડ્રેગ કરો.
- હવે માઉસ બટન છોડી દો.

સેલને ફોર્મેટ કરવો (Formatting a Cell)

આપણે સેલના ગુણધર્મ બદલી શકીએ છીએ. આપણે સેલને રંગ લાગુ પાડી શકીએ, સેલની અંદર ઇમેજ ઉમેરી શકીએ, સેલને કિનારી આપી શકીએ અને સેલની અંદર રહેલ માહિતીના ફોન્ટ પણ બદલી શકીએ છીએ. આપણે સેલની માહિતીને વિવિધ ફોર્મેટ (format) લાગુ પાડી શકીએ જેમ કે તારીખ અને સંખ્યાનું ફોર્મેટ બદલવું. એટલે કે, સંખ્યાને, અંદર રહેલ માહિતીને, એલાઈનમેન્ટ, સેલની કિનારી અને સેલની પશ્ચાદ્ભૂમિને ફોર્મેટ કરી શકાય છે. આ કાર્ય માટે Format cells ડાયલોગ બોક્સ અથવા ફોર્મેટિંગ ટૂલબારનો ઉપયોગ કરી શકીએ. આકૃતિ 6.14માં ફોર્મેટિંગ ટૂલબાર દર્શાવ્યું છે.



આકૃતિ 6.14 : લેબલ સાથેનું ફોર્મેટિંગ ટૂલબાર

આકૃતિ 6.14માં દર્શાવેલાં લેબલનું વર્ણન કોષ્ટક 6.4માં આપેલું છે.

લેબલ	વર્ણન
1	ઢબ (style) લાગુ પાડો
2	ફોન્ટ
3	ફોન્ટ સાઈઝ (ફોન્ટનું કદ)
4	ઘાટા, ત્રાંસા (italic) અને શબ્દ નીચે લીટી દોરવી
5	ગોઠવણ (એલાઈમેન્ટ) : ડાબી બાજુ, મધ્યમાં, જમણી બાજુ, જસ્ટીફાઇ
6	પસંદ કરેલાં સેલ ભેળવવાં (મર્જ કરવા)
7	ચલણ
8	ટકા
9	દશાંશ ચિહ્ન ઉમેરવું / રદ કરવું
10	ઇન્ડેન્ટ વધારવું / ઘટાડવું
11	સેલની કિનારીનું ફોર્મેટ
12	સેલની પશ્ચાદ્ભૂમિનો રંગ
13	સેલમાં ફોન્ટનો રંગ
14	ફોર્મેટ ન કરેલી સેલની કિનારી ઉમેરવી / રદ કરવી

ફોર્મેટ 6.4 : ફોર્મેટિંગ ટૂલબાર ઉપરના વિકલ્પો

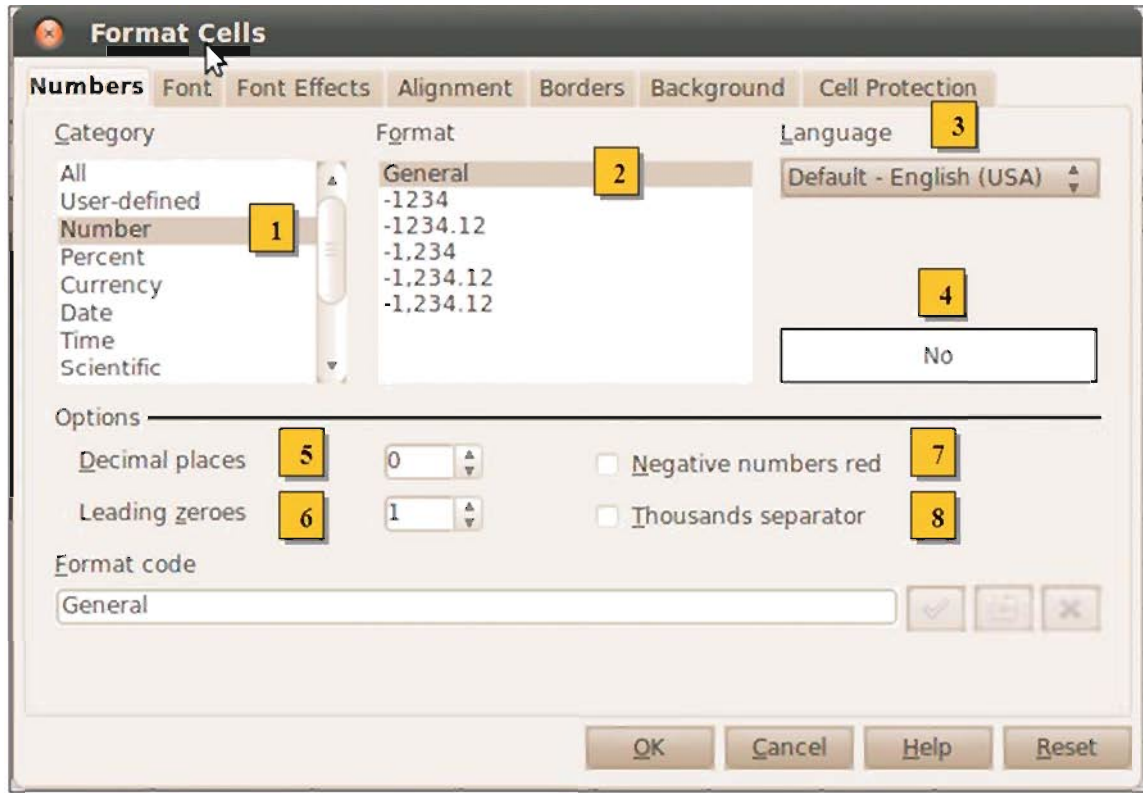
ચાલો, હવે આપણે ફોર્મેટિંગની કેટલીક ઢબ જોઈએ. તે પછી તમે ઉપર જણાવેલી વિવિધ ફોર્મેટિંગ સ્ટાઇલનો વર્કશીટના કેટલાક સેલ ઉપર પ્રયોગ કરી જુઓ.

નંબર ફોર્મેટ (Number Format)

નંબર ફોર્મેટ સેલમાં રહેલી સંખ્યાને અસર કરે છે. આ નંબર ફોર્મેટ કોઈ એક સેલ, એક સાથે પસંદ કરેલા અનેક સેલ અને સેલના વિસ્તારને લાગુ પાડી શકાય છે. આકૃતિ 6.15માં દર્શાવ્યા પ્રમાણે Format cells પસંદ કરીને Numbers ટેબ ખોલો અને સ્કીન ઉપર પ્રદર્શિત ડાપલોંગ બોક્સમાંથી તમારી પસંદગી પ્રમાણે ફોર્મેટ સેટ કરો. આકૃતિ 6.15માં જે લેબલ આપેલાં છે તેની સમજ કોષ્ટક 6.5માં આપેલી છે.

લેબલ	વર્ણન
1.	યોગ્ય ફોર્મેટની પસંદગી જેમ કે Numbers, dates વગેરે.
2.	પસંદ કરેલાં ફોર્મેટનો લાક્ષણિક દેખાવ
3.	પ્રાદેશિક સેટિંગ્સ
4.	પસંદ કરેલા સેલનું પ્રિવ્યુ, હજાર માટે સેપરેટર (અલગ પાડવા માટેનું ચિહ્ન) ઉમેરવું
5.	દશાંશ ચિહ્ન પછી અંકની સંખ્યા
6.	શરૂઆતનાં શૂન્યની સંખ્યા
7.	ઋણ સંખ્યાને લાલ દર્શાવવી
8.	હજારમાં અલગ પાડવું

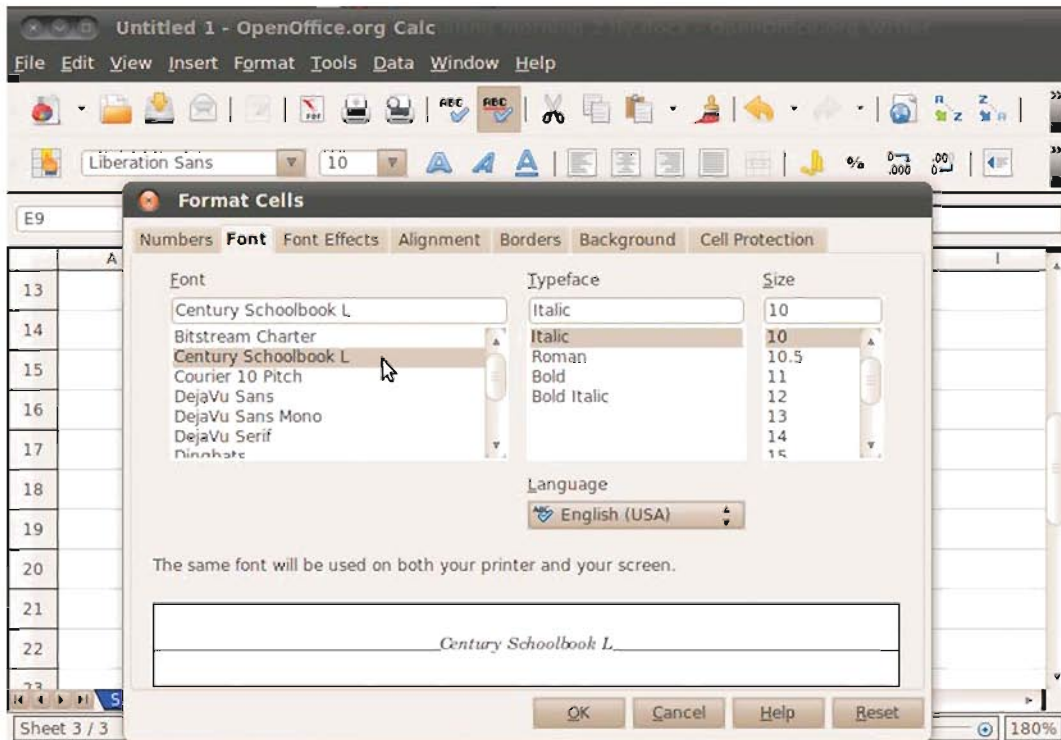
કોષ્ટક 6.5 : સેલ માટે Number ફોર્મેટિંગના વિકલ્પો



આકૃતિ 6.15 : સેલ માટે Numbers ફોર્મેટિંગના વિકલ્પો

સંખ્યા ધરાવતા સેલ માટેના Number ફોર્મેટિંગના વિકલ્પો આકૃતિ 6.15માં દર્શાવ્યા છે. તારીખ, ચલણ કે ટકાનું ફોર્મેટ બદલવા માટે યોગ્ય પસંદ કરવી જરૂરી છે.

Format Cells ડાયલોગ બોક્સના બીજા વિકલ્પ **Font** ટેબ દ્વારા તમે ફોન્ટ બદલી શકો છો. આકૃતિ 6.16માં દર્શાવ્યા પ્રમાણે **Font Effects** ટેબ વડે ફોન્ટની અસર બદલી શકાય છે.



આકૃતિ 6.16 : ફોન્ટ બદલવા

તમે કદાચ આમાંની કેટલીક ફોન્ટની અસરો લાગુ પાડી પણ હોય જેમ કે ઘાટા (બોલ્ડ), ત્રાંસા (ઇટાલિક), શબ્દની નીચે લીટી (અન્ડરલાઇન) અથવા ફોન્ટનું કદ. આમાંની કેટલીક અસર ફોર્મેટિંગ ટૂલબાર ઉપર આઈકોન તરીકે આપેલી છે. ફક્ત યોગ્ય આઈકોન પસંદ કરીને આ અસરનો સીધો અમલ કરી શકાય છે.

ડેટાની ક્રમબદ્ધ ગોઠવણી અને નિર્ણયન (ડેટા સોર્ટિંગ અને ફિલ્ટરિંગ - Data Sorting and Filtering)

જો કોઈ ચોક્કસ ક્રમમાં ડેટા ગોઠવાયેલા હોય તો તેને સમજવા અને વાપરવા સરળ બને છે. ડેટાની કોઈ ચોક્કસ ક્રમમાં ગોઠવણીને સોર્ટિંગ (sorting) કહેવામાં આવે છે. ડેટાને ચઢતા કે ઊતરતા ક્રમમાં ગોઠવવાની પ્રક્રિયા કેલ્સીમાં ઉપલબ્ધ છે. જો ડેટા એક જ કોલમ કે રોમાં હોય તો તે તમારી ઇચ્છિત રીતે સોર્ટ થશે. પણ જો ડેટા એક કરતાં વધારે રો કે કોલમમાં વિસ્તરેલો હોય તો બધી જ કોલમ અને રો બદલાશે અને સોર્ટિંગ થશે.

ઉદાહરણ તરીકે આપણે વિદ્યાર્થીઓના ગુણ લઈએ. કયા વિદ્યાર્થીએ સૌથી વધારે કુલ ગુણ મેળવ્યા છે તે શોધી કાઢવા માટે અને મેરિટ લિસ્ટ તૈયાર કરવા માટે આપણે ઊતરતા ક્રમમાં (સૌથી મોટી કિંમત પહેલી આવશે) સોર્ટિંગ કરવું પડશે. આ અજમાયશ કરવા માટે એક નવી સ્પ્રેડશીટ ખોલો અને આકૃતિ 6.17માં દર્શાવ્યા પ્રમાણે ડેટા દાખલ કરો. તે પછી યોગ્ય નામ આપી ફાઈલનો સંગ્રહ કરો.

	A	B	C	D	E	F	G	H
1	No	Name	Test 1	Test 2	Total			
2	1	Sita	56	44	100			
3	2	Mita	60	55	115			
4	3	Rita	65	70	135			
5	4	Gita	40	55	95			
6	5	Nita	30	35	65			
7								
8								
9								
10								

આકૃતિ 6.17 : વર્કશીટમાં દાખલ કરેલી માહિતી

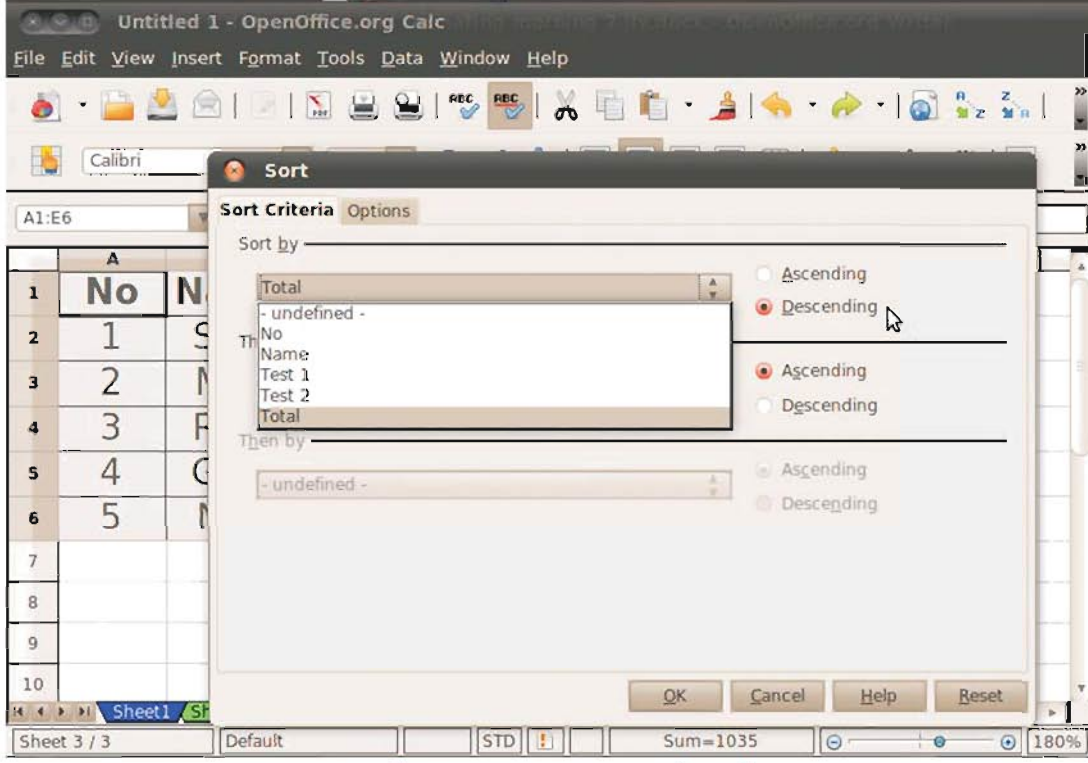
આકૃતિ 6.17માં ઉપલબ્ધ ડેટાને કોલમ Eની કિંમત, એટલે કે **'Total'** ઉપર સોર્ટ કરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- જે કોલમના ડેટા ઉપર સોર્ટ કરવાની જરૂર હોય તેનાં કોઈ પણ સેલ ઉપર ક્લિક કરો.
- આકૃતિ 6.8માં દર્શાવેલું **Formatting** ટૂલબાર ઉપરના **Sort descending** બટન ઉપર ક્લિક કરો.



આકૃતિ 6.18 : ફોર્મેટિંગ ટૂલબારનો ઉપયોગ કરીને સોર્ટિંગ

આ જ ક્રિયા બીજી રીતે પણ થઈ શકે. આ માટે તમારે ફક્ત **Data → Sort** પસંદ કરવાનું છે. આથી એક ડાયલોગ બોક્સ પ્રદર્શિત થશે. ડાયલોગ બોક્સના sort criteriaમાં **Descending** (ઊતરતો ક્રમ) પસંદ કરો. આકૃતિ 6.19માં sort ડાયલોગ બોક્સનો એક લાક્ષણિક દેખાવ આપેલો છે.



આકૃતિ 6.19 : Sort ડાયલોગ બોક્સ

ક્રમબદ્ધ ગોઠવ્યા પછી તમે જોઈ શકશો કે કોલમમાં કુલ ગુણ **135** પહેલાં સ્થાન ઉપર આવેલ છે. સ્પષ્ટ રીતે આ ગુણ **Rita** નામના વિદ્યાર્થીએ મેળવેલા છે અને આથી વિદ્યાર્થીઓના નામની કોલમમાં તેનું નામ પ્રથમ આવવું જોઈએ. જો તમે પરિણામ તરફ નજર કરશો તો જણાશે કે રીટાના કુલ ગુણ સાથે તેનો અન્ય ડેટા પણ પહેલી હરોળમાં ખસેડાયો છે. આનો અર્થ એ થાય કે તમે કોઈ કોલમ ઉપર સોર્ટ કરો છો ત્યારે આકૃતિ 6.20માં દર્શાવ્યા પ્રમાણે ઘણી રો પણ સ્થાન બદલશે. આકૃતિ 6.20માં બદલાયેલાં રેકૉર્ડ(ડેટા)ના ક્રમની નોંધ કરો.

	A	B	C	D	E	F	G	H
	No	Name	Test 1	Test 2	Total			
2	3	Rita	65	70	135			
3	2	Mita	60	55	115			
4	1	Sita	56	44	100			
5	4	Gita	40	55	95			
6	5	Nita	30	35	65			
7								
8								
9								
10								

આકૃતિ 6.20 : ઊતરતા ક્રમમાં ગોઠવાયેલો ડેટા

તમે સંખ્યા ઉપર જ ફક્ત સોર્ટ નથી કરી શકતા પણ સોર્ટિંગ મૂલ્યાંકરો આધારિત પણ કરી શકો છો. અંગ્રેજી ભાષાના શબ્દકોશનું ઉદાહરણ લો કે જેમાં શબ્દો કક્કાવાર ગોઠવાયેલાં હોય છે. આપણે હાલમાં જે વિદ્યાર્થીઓના ડેટા ઉપર પ્રયોગ કરી રહ્યા છીએ તેની ઉપર નજર ફેરવો અને તે ડેટાને વિદ્યાર્થીઓનાં નામના ક્રમમાં ગોઠવવાનો પ્રયત્ન કરો. ઉપર જણાવેલ ડેટાને વિદ્યાર્થીઓના નામના ચઢતા ક્રમમાં ગોઠવવા (સોર્ટ કરવા) નીચે જણાવ્યા પ્રમાણે કાર્ય કરો :

- પસંદ કરો : **Data → Sort**
- હવે કોલમનું નામ **Name** આપો.
- સોર્ટ ક્રમ તપાસી જુઓ, પૂર્વનિર્ધારિત રીતે તે **Ascending** છે.

તમે ડેટા બે અથવા વધારે ક્ષેત્ર (ફિલ્ડ ફિલ્ડ) ઉપર પણ સોર્ટ કરી શકો છો. એવી પરિસ્થિતિ વિચારો કે તમારી પાસે વિદ્યાર્થીઓનાં પૂર્ણ નામ, તેની જગ્યા અને તેનાં ગુણ જેવી માહિતી સાથેના વિદ્યાર્થીઓની યાદી છે. તમે ડેટાને પ્રથમ જગ્યા (location) અને પછી તેના નામ (name) ઉપર સોર્ટ કરવા ઇચ્છો છો. નીચે આપેલું ઉદાહરણ લો.

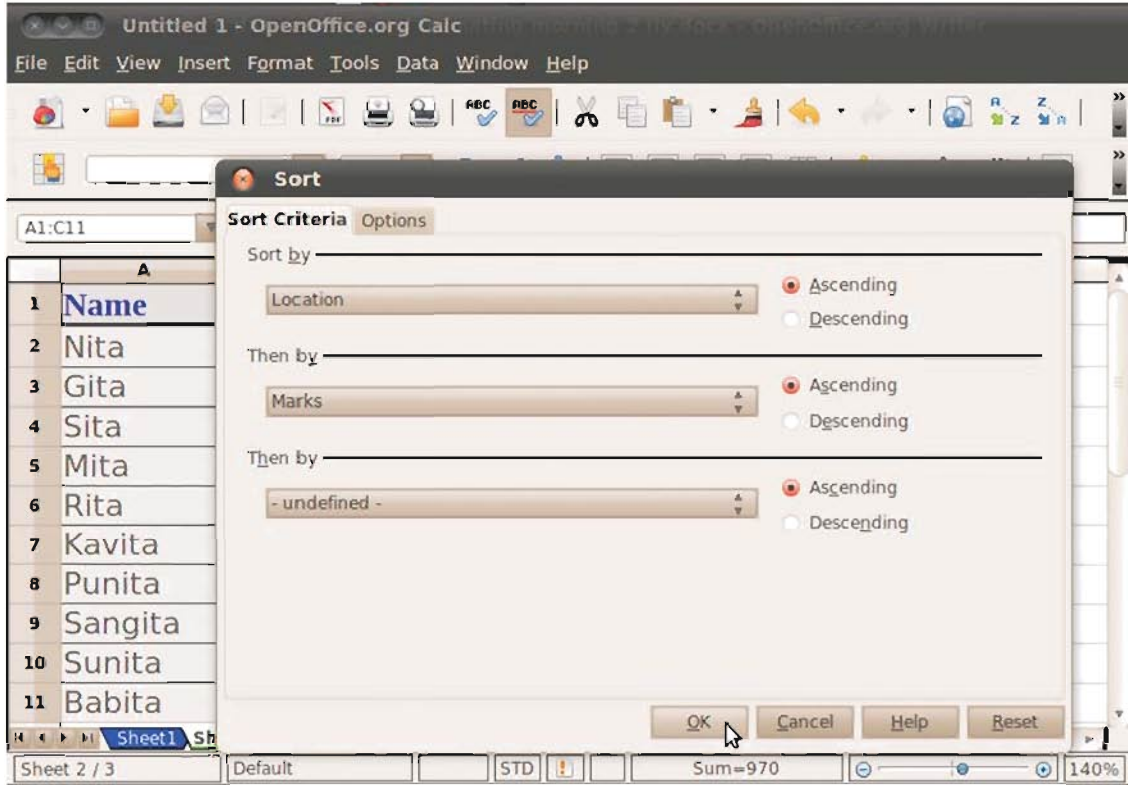
1. કોષ્ટક 6.6માં આપેલો ડેટા કેલ્સી વર્કશીટમાં દાખલ કરો.

Name	Location	Marks
Nita	Vallabh vidyanagar	65
Gita	Anand	95
Sita	Baroda	100
Mita	Ahmedabad	115
Rita	Gandhinagar	135
Kavita	Vallabh vidyanagar	110
Punita	Ahmedabad	105
Sangita	Anand	85
Sunita	Ahmedabad	70
Babita	Anand	90

કોષ્ટક 6.6 : સોર્ટ કરવા માટે નમૂનાનો ડેટા

2. શીર્ષકો સાથે ડેટા પસંદ કરો.
3. પસંદ કરો : **Data → Sort**
4. **Sort by** ફિલ્ડમાં **"Location"** પસંદ કરો.
5. **Then by** ફિલ્ડમાં **"Marks"** પસંદ કરો.
6. હવે **OK** બટન દબાવો.

આ બધી ક્રિયાઓ આકૃતિ 6.21માં દર્શાવી છે.



આકૃતિ 6.21 : સોર્ટના વિકલ્પો

આ બધાં પગલાંઓનો અમલ કરવાથી પસંદ કરેલો ડેટા પ્રથમ location (જગ્યા) અને પછી marks (ગુણ) ઉપર સોર્ટ થશે (ગોઠવાશે). આનો અર્થ એ થાય કે Ahmedabad ના બધા વિદ્યાર્થીઓની યાદી તેણે મેળવેલા ગુણના ચઢતા ક્રમમાં (સૌથી ઓછા ગુણ ધરાવતો વિદ્યાર્થી પ્રથમ) મળશે. અહીં location ચઢતા ક્રમમાં સોર્ટ કરતા હોવાથી Ahmedabad સૌપ્રથમ અને Vallabh Vidyanagar અંતમાં આવશે. સોર્ટ કરેલો ડેટા આકૃતિ 6.22માં દર્શાવ્યા જેવો દેખાશે. જો જરૂર હોય તો તમે ત્રીજા લેવલનું સોર્ટ પણ ઉમેરી શકો છો.

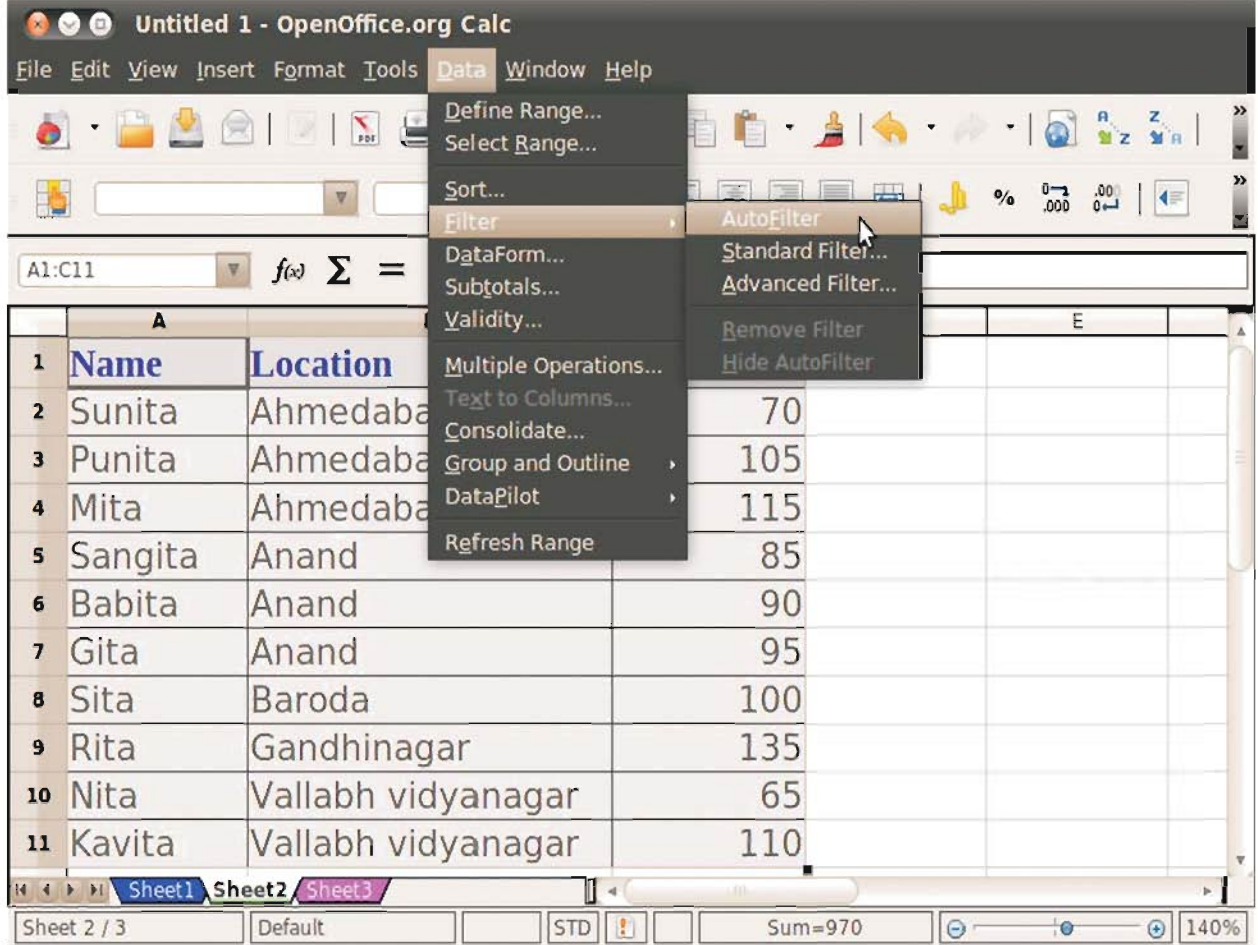
	A	B	C	D	E
	Name	Location	Marks		
1	Sunita	Ahmedabad	70		
2	Punita	Ahmedabad	105		
3	Mita	Ahmedabad	115		
4	Sangita	Anand	85		
5	Babita	Anand	90		
6	Gita	Anand	95		
7	Sita	Baroda	100		
8	Rita	Gandhinagar	135		
9	Nita	Vallabh vidyanagar	65		
10	Kavita	Vallabh vidyanagar	110		

આકૃતિ 6.22 : ગોઠવણી પછીનો ડેટા (સોર્ટેડ ડેટા)

ડેટા નિસ્યંદન (Data filtering)

ડેટા ફિલ્ટરિંગ આપણને ડેટા ગાળીને બિનજરૂરી ડેટા દૂર કરવામાં મદદ કરે છે અને તમે જે ઇચ્છો તે જ ડેટા રજૂ કરે છે. આપણે હમણાં જ અગાઉના વિભાગમાં સોર્ટ કરેલો ડેટા કે જેમાં વિદ્યાર્થીઓનાં નામ અને જગ્યા વિશે હતો. તો હો. ચાલો આપણે ડેટા ઉપર એક નિસ્યંદક (ફિલ્ટર-filter) ઉમેરીએ. ધારો કે આપણે એ વિદ્યાર્થીઓની માહિતી જોવા ઇચ્છીએ છીએ કે જેમણે 100 અથવા 100 કરતાં વધારે ગુણ મેળવ્યા હોય. આ કાર્ય કરવા માટે ઓટો ફિલ્ટર આપણે નીચે પ્રમાણે ઉમેરી શકીએ :

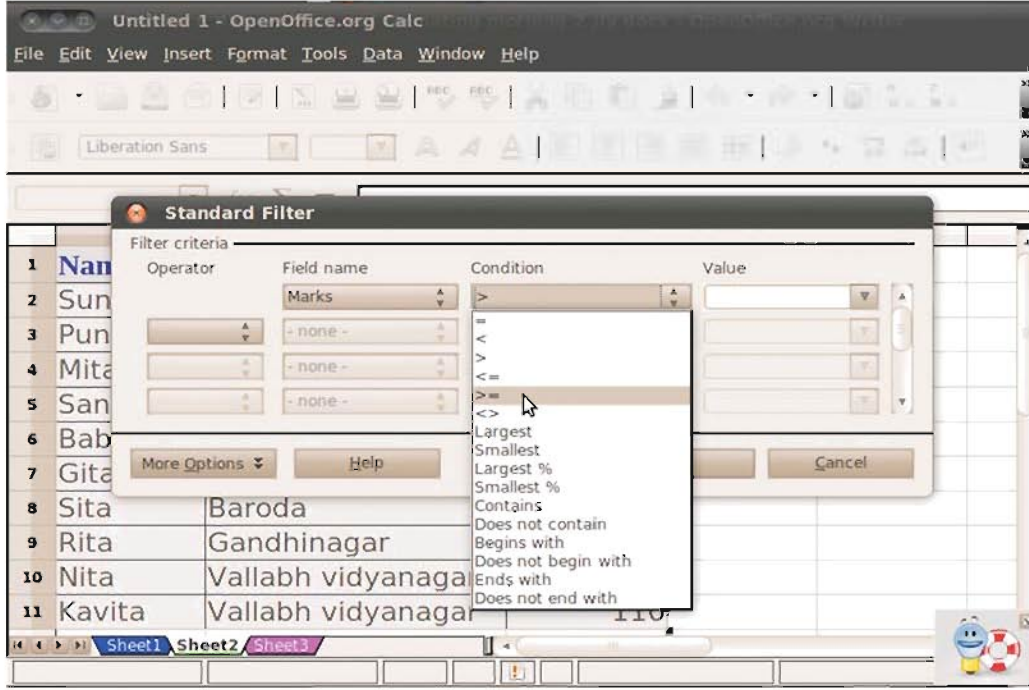
1. આકૃતિ 6.23માં દર્શાવ્યા પ્રમાણે **Data → AutoFilter** પસંદ કરો. તમે હયાત વર્કશીટ કે જેમાં પહેલાં તમે ડેટા દાખલ કરેલો છે તે પણ વાપરી શકો છો.



આકૃતિ 6.23 : પસંદ કરેલા ડેટા ઉપર ફિલ્ટર લાગુ કરવા

2. દરેક કોલમના મથાળે તમે ડ્રોપડાઉન એરો (dropdown arrows) જોઈ શકશો.
3. ક્લિક કરી કોઈ ચોક્કસ કોલમ(અહીં Marks)નાં તીર(એરો)ને પકડી રાખો અને કિંમત પસંદ કરો. જેમકે, તમે અગાઉ દાખલ કરેલી કિંમતો અથવા વાક્યપદ (phrases) જેમ કે પહેલાં 10 શ્રેષ્ઠ વગેરે. જો તમારે કોઈ વિશિષ્ટ વિકલ્પ જોઈતો હોય તો તમે standard filter પસંદ કરી આકૃતિ 6.24માં દર્શાવ્યા પ્રમાણે તેમાંથી કિંમતો પસંદ કરો.
4. તમે જોઈ શકશો કે સ્પ્રેડશીટમાં જે વિદ્યાર્થીના ગુણ 100 કરતાં ઓછા હતા તે ગાળીને દૂર કરી દીધા છે. આકૃતિ 6.25માં દર્શાવ્યા પ્રમાણે તીરનો બદલાયેલો રંગ કે જે કેટલોક ડેટા ગાળીને દૂર કર્યો છે તે સૂચવે છે, તેનું કદાચ તમે નિરીક્ષણ કર્યું હશે.

ફરી પસંદ કરો : **Data → Filter → AutoFilter** જેથી ફિલ્ટરનું કાર્ય બંધ થાય.



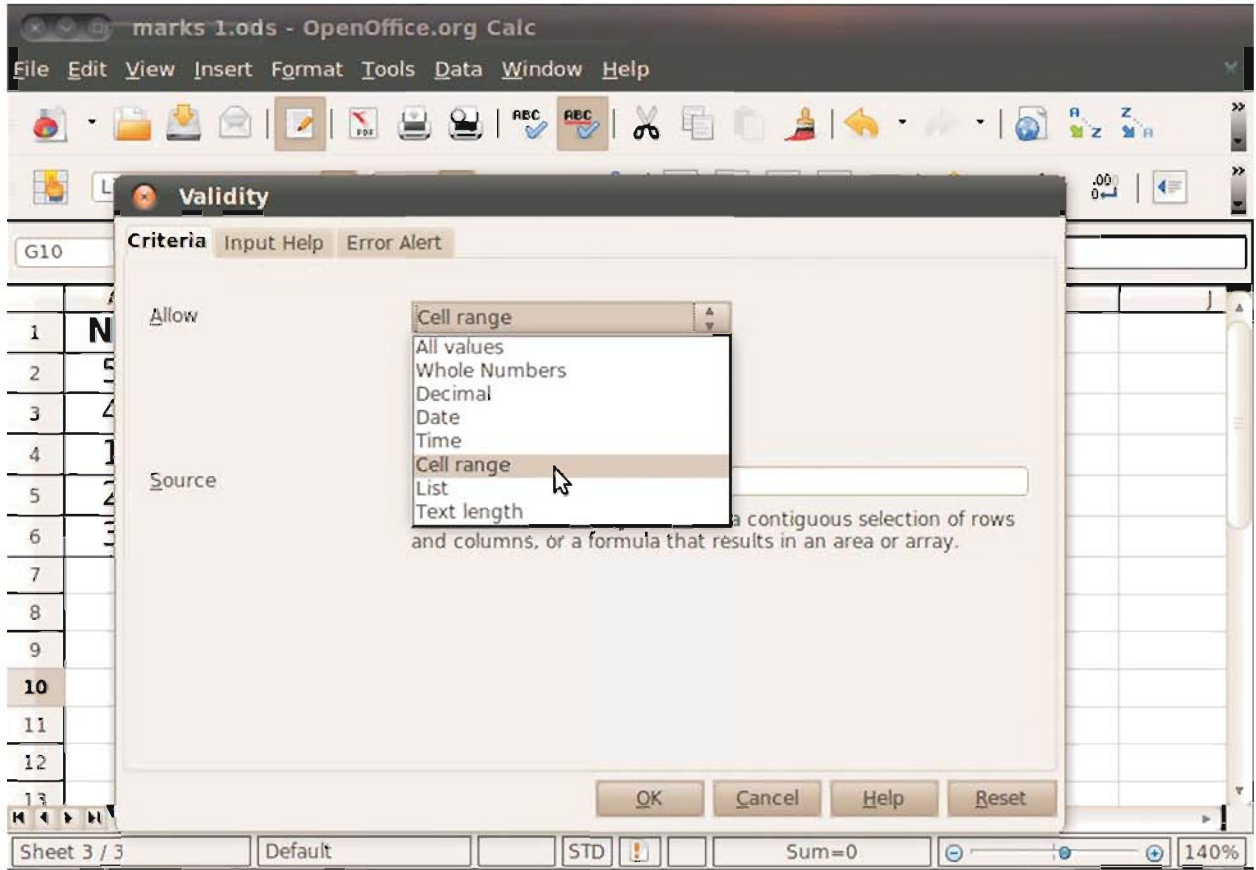
આકૃતિ 6.24 : Standard filter ડાયલોગ બોક્સ

	A	B	C	D	E
1	Name	Location	Marks		
3	Punita	Ahmedabad	105		
4	Mita	Ahmedabad	115		
8	Sita	Baroda	100		
9	Rita	Gandhinagar	135		
11	Kavita	Vallabh vidyanagar	110		
12					
13					
14					
15					
16					
17					

આકૃતિ 6.25 : ગાળેલો (ફિલ્ટર્ડ) ડેટા

ડેટા ચકાસણી (Data Validations)

અગાઉ જણાવ્યું હતું તે પ્રમાણે સોડશીટ પેકેજનો ઉપયોગ માહિતી વિશ્લેષણ જેવાં કે સરવેયું તૈયાર કરવું અને નાણાંકીય વિશ્લેષણ, મેરિટ યાદી તૈયાર કરવી અને વિદ્યાર્થીઓના પરિણામના વિશ્લેષણમાં થાય છે. આ કિસ્સામાં જો તમે માન્ય (valid) માહિતી આપી હશે તો જ તમને આપેલાં સૂત્રો અને વિધેયો સાચું પરિણામ આપશે. જો કે, સિસ્ટમના ઉપયોગકર્તા હંમેશા માન્ય ડેટા દાખલ કરશે તેની કોઈ ખાતરી આપી શકતું નથી. ડેટા દાખલ કરતાં સમયે ભૂલો રોકવા (ટાળવા) માટે આપણે ડેટા વેલિડેશન (data validations)નો ઉપયોગ કરી શકીએ છીએ. સેલમાં રહેલી માહિતી સાચી છે (માન્ય છે) તેની ખાતરી કરવા માટે સેલ પસંદ કરો અને **Data → Validity** આદેશ આપો. આથી આકૃતિ 6.26માં દર્શાવ્યા પ્રમાણે Validity ડાયલોગ બોક્સ ખૂલશે.



આકૃતિ 6.26 : Validityના વિકલ્પો

જે ડેટા તમે ઉપયોગકર્તા પાસેથી મેળવો છો તેના માટે હવે તમે અલગ ધોરણ સેટ કરી શકો છો. અમાન્ય ડેટાને દાખલ થતો રોકવા માટે આપણે અહીં કેટલીક આવશ્યકતાઓ (શરતો) સેટ કરી શકીએ. આ આવશ્યકતાઓમાં સેલ વિસ્તાર, બધી કિમતો, ફક્ત પૂર્ણાંક સંખ્યાઓ દાખલ કરવી, ફક્ત તારીખ જ માન્ય રાખવી વગેરેનો સમાવેશ થાય છે, જેના માટે Cell range, All values, ફક્ત Whole numbers અને Date તરીકે વિકલ્પો હોય છે. આ પ્રકારનાં સેટિંગ આપણે **Criteria** ટેબ વડે આકૃતિ 6.26માં દર્શાવ્યા પ્રમાણે કરી શકીએ છીએ. બીજો ટેબ **Input Help** છે જેના વડે ડેટા દાખલ કરતાં સમયે આપણે ઉપયોગકર્તાને સહાય પૂરી પાડી શકીએ છીએ.

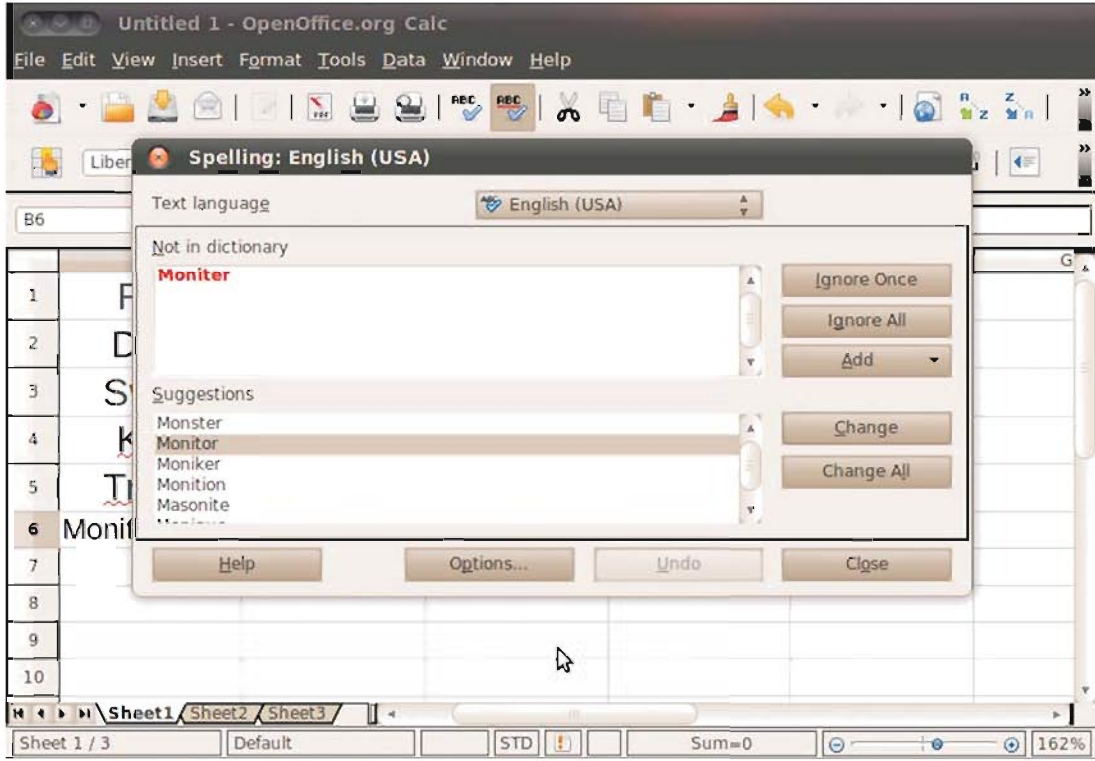
આ ઉપરાંત અન્ય એક **Error Alert** ટેબ હોય છે કે જો કંઈ ભૂલ થાય તો ભૂલ માટે એક ચેતવણી (એલર્ટ) આપે છે. અહીં એ નોંધવું જોઈએ કે જો તમે **Delete Contents** વિન્ડોમાંથી **Format** અથવા **Delete All** પસંદ કરશો તો તમે સેટ કરેલા Validity વિકલ્પો દૂર કરવામાં આવશે.

કેલ્સીમાં અન્ય પ્રક્રિયાઓ (Other operations in Calc)

કેલ્સી અન્ય અનેક ઉપયોગી પ્રક્રિયાઓ પૂરી પાડે છે. જેમ કે, અંગ્રેજી શબ્દોની જોડણી(સ્પેલિંગ)ની ચકાસણી, શાબ્દિક લખાણ શોધવું અને બદલવું, વર્કશીટ છાપવી વગેરે. આમાંના કેટલાક વિકલ્પોની અહીં આ વિભાગમાં ચર્ચા કરી છે.

જોડણીની ચકાસણી (Spelling check)

કેલ્સીની વર્કશીટમાં માહિતી ટાઈપ કરતી વખતે ઘણીવાર ઉપયોગકર્તા અંગ્રેજીના સ્પેલિંગમાં ભૂલો કરે છે. કેલ્સી આપણને જોડણી (સ્પેલિંગ) ચકાસવાની સુવિધા આપે છે. જો તમે ટૂલબાર ઉપરનું Spelling check બટન (કે જે ટોગલ બટન છે અને તેને ઓન કે ઓફ કરી શકાય છે) ઓન રાખેલું હશે તો ખોટી રીતે લખાયેલા શબ્દો લાલ રેખાથી રેખાંકિત થશે. જ્યારે કંઈક ખોટો શબ્દ (સ્પેલિંગમાં ભૂલ) ટાઈપ કરશો તો તે શબ્દની નીચે લાલ લીટી સાથે તે શબ્દ હાઈલાઈટ થશે. તમારે ફક્ત તેના ઉપર રાઈટ ક્લિક કરવાની જરૂર હોય છે અને આપેલી પસંદગીઓમાંથી યોગ્ય વિકલ્પ પસંદ કરો. જોડણી ચકાસવા માટે કેલ્સી શબ્દકોશનો ઉપયોગ કરે છે. જો તમે દાખલ કરેલો શબ્દ તે જ સ્પેલિંગ સાથે શબ્દકોશમાં મળી જાય તો તે શબ્દ સાચો ગણવામાં આવે છે. આકૃતિ 6.27માં જોડણીની ચકાસણી અને સુધાર માટેના શક્ય વિકલ્પ દર્શાવ્યા છે.



આકૃતિ 6.27 : જોડણી ચકાસણીના વિકલ્પો

અમુક સમયે એવું પણ બને કે શબ્દ સાચો હોય જેમ કે તમારું નામ, પણ તે શબ્દ પ્રમાણભૂત શબ્દકોશમાં ન હોય. આ કિસ્સામાં તમારા નામ નીચે પણ લાલ રેખા હોઈ શકે ! આવી પરિસ્થિતિ ટાળવા માટે તમારું નામ "Add to dictionary" વિકલ્પ વડે શબ્દકોશમાં ઉમેરી શકો. તે પછી તમારું નામ અમાન્ય શબ્દ તરીકે હાઈલાઈટ થશે નહીં. જોડણી ચકાસણી કરવા માટેનો એક સરળ માર્ગ છે માત્ર ફંક્શન કી F7ને દબાવવાનો. આપમેળે જોડણી ચકાસવાનું ટાળવા માટે તમે આ પ્રમાણે પસંદ કરી શકો : **Format → Cells → Font language=[None]**

શોધવું અને બદલવું (Find and Replace)

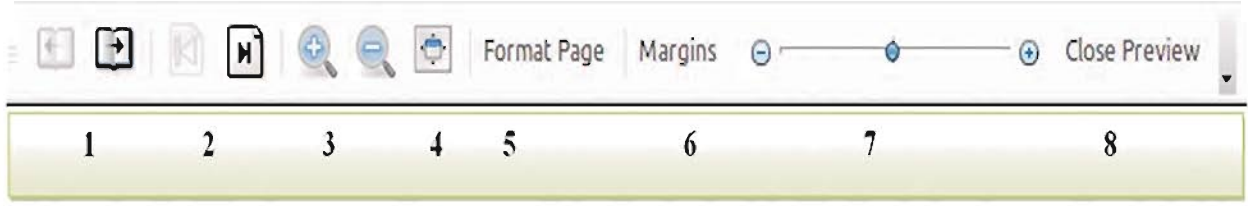
Find and Replace આદેશ વર્કશીટમાં ખૂબ ઝડપથી માહિતી (ડેટા) શોધવામાં અને સેલમાંનું શાબ્દિક લખાણ કે સંખ્યા સંપૂર્ણપણે અથવા આંશિક રીતે બદલવા માટે વપરાય છે. આ કાર્ય કરવા માટે નીચે પ્રમાણે પગલાં ભરો :

- પસંદ કરો : **Edit → Find & Replace**
- Find & Replace ડાયલોગ બોક્સમાં પૂછવામાં આવેલી જરૂરી ક્રિયાઓ કરો.

મુદ્રણ (Printing)

વર્કશીટના મોટા કદને કારણે વર્કશીટ છાપવા માટે (પ્રિન્ટિંગ માટે) ખાસ કાળજી જરૂરી છે. શબ્દ-પ્રક્રિયક (વર્ડ પ્રોસેસર) સોફ્ટવેરની જેમ સ્પ્રેડશીટ વિનિયોગના દસ્તાવેજનું માપ નિશ્ચિત નથી. તમે અનેક આડી હરોળ (રો) અને ઊભી હરોળ (કોલમ) વડે બનાવેલો દસ્તાવેજ એક પાનામાં બંધબેસતો ન પણ આવે. આ ઉપરાંત કેટલીક માહિતી કપાઈ જાય એમ પણ બને. વધુમાં, આખી વર્કશીટ છાપવી જરૂરી પણ નથી હોતી. આથી, કોઈ પણ દસ્તાવેજ છાપતાં પહેલાં, print preview (મુદ્રણ પૂર્વેનું અવલોકન) વિકલ્પ વાપરવો સલાહભર્યું છે. print preview વિકલ્પ પાનું પૂરું થાય તે નિશાની (પેઈજ બ્રેક) અને હાંસિયા (માર્જિન) સાથે સ્પ્રેડશીટનો દેખાવ પૂરો પાડે છે, વર્કશીટના પૂર્વાવલોકન (પ્રિવ્યૂ) માટે નીચે પ્રમાણે પગલાં ભરો :

- પસંદ કરો : **File → Page Preview**
- આથી કમ્પ્યુટર સ્ક્રીનના ઉપરના ભાગમાં આકૃતિ 6.28માં દર્શાવ્યા પ્રમાણેનો એક બાર (bar) જોવા મળશે. તે દસ્તાવેજના પૂર્વાવલોકન માટે વપરાશકર્તા ઉપયોગમાં લઈ શકે તે ક્રિયાઓ દર્શાવે છે.



આકૃતિ 6.28 : પૂર્વાવલોકન (પ્રિવ્યુ) બાર

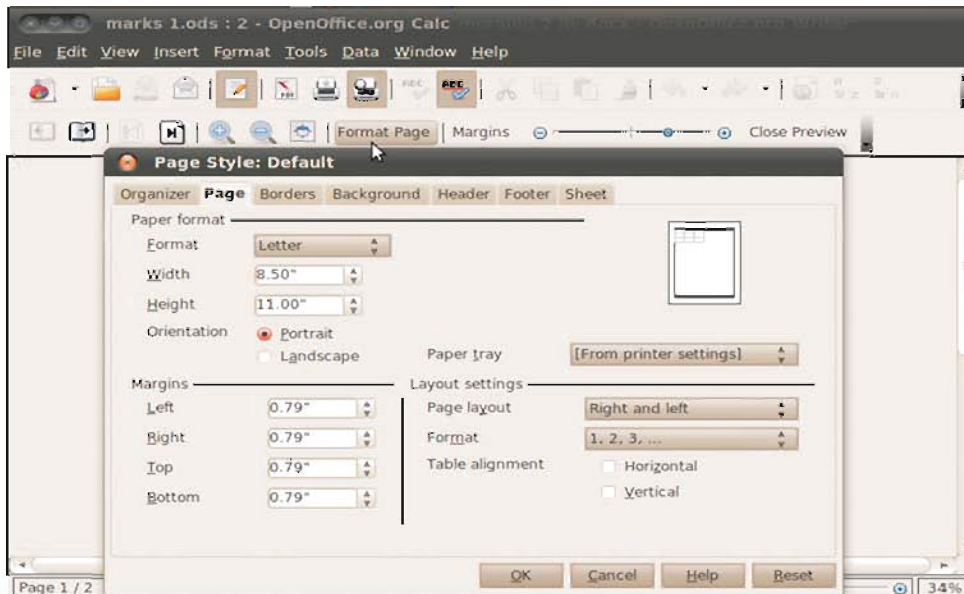
યોગ્ય આઈકોન પસંદ કરી આપણે જરૂરી પ્રક્રિયાઓ કરી શકીએ છીએ. આકૃતિ 6.28માં દર્શાવેલાં લેબલ જુદી જુદી ક્રિયાઓ માટે છે, તેની યાદી કોષ્ટક 6.7માં આપેલી છે.

લેબલ	વર્ણન
1	પાનાં બદલવાં
2	છેલ્લા પાના પર પહોંચવા
3	ઝૂમ ટૂલ
4	આખા સ્ક્રીન પરનો દેખાવ
5	પાનાના ફોર્મેટિંગ માટેનું ડાયલોગ બોક્સ ખોલવા માટે
6	હાંસિયા દર્શાવવા
7	સ્કેલિંગ
8	પ્રિવ્યુ બંધ કરવા

કોષ્ટક 6.7 : પૂર્વાવલોકન વિકલ્પો

પાનાનું પૂર્વાવલોકન પાનાના ફોર્મેટિંગની કિંમતો સેટ કરવા માટે પણ વપરાય છે. પાનાનું ફોર્મેટિંગ (page formatting) કરવા માટે નીચે પ્રમાણે પગલાં ભરો :

- પસંદ કરો : **File → Page Preview**
- સ્ક્રીન ઉપર પ્રદર્શિત આડા ડાયલોગ બોક્સમાંથી **Format Page** બટન દબાવો.
- આકૃતિ 6.29માં દર્શાવ્યા પ્રમાણે **Page Style** નામના ડાયલોગ બોક્સમાં **Page** લખેલા વિભાગને પસંદ કરો.
- જરૂરી પસંદગીઓ પસંદ કરી OK બટન દબાવી પરિવર્તનોનો સંગ્રહ કરો.

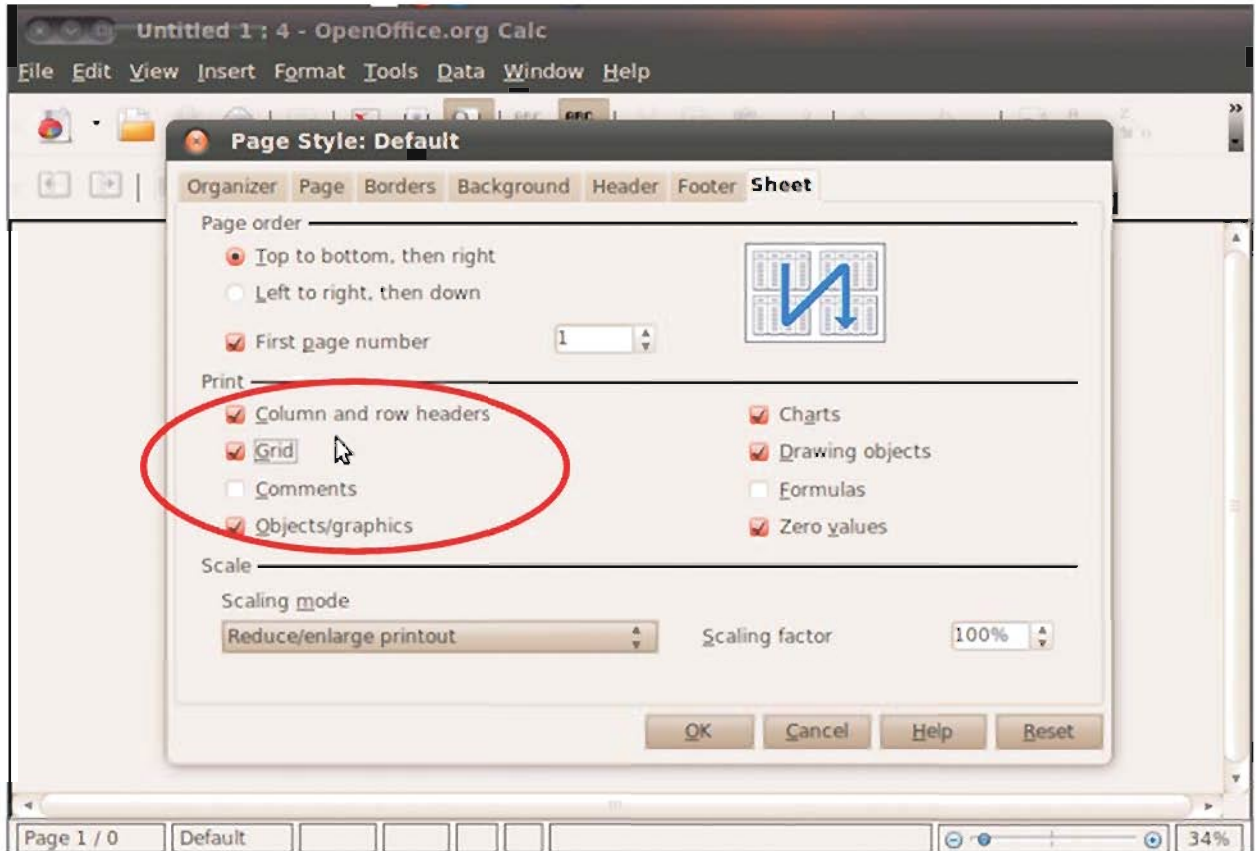


આકૃતિ 6.29 : Page formattingના વિકલ્પો

તમે પેપરનું માપ (**Format**) A4, લીગલ અથવા કોઈ પણ ઇચ્છિત પાનાના કદની પસંદગી કરી શકો છો. તમે **Margins** વડે ડાબી બાજુ, જમણી બાજુ, ઉપર અને નીચે માર્જિન (હાંસિયા) સેટ કરી શકો છો, આ વિકલ્પો ઉપરાંત તમે અન્ય Page formattingના વિકલ્પોના પ્રયોગો કરી શકો : **Paper Tray** વિકલ્પ કે જેમાં ઉપરની ટ્રેમાંથી કે નીચેની ટ્રેમાંથી તમારું પાનું પ્રિન્ટરમાં આવશે તે જણાવવાનું હોય છે અને **Page Layout** વિકલ્પનો ઉપયોગ એવા સમયે થાય છે જ્યારે એક ભૌતિક પાના ઉપર તમે એક કરતાં વધારે પાનાં છાપતાં હો. Formatની પસંદગી આડું પાનું કે ઊભું પાનું જેવું **Orientation** જણાવવા માટે પણ થાય છે.

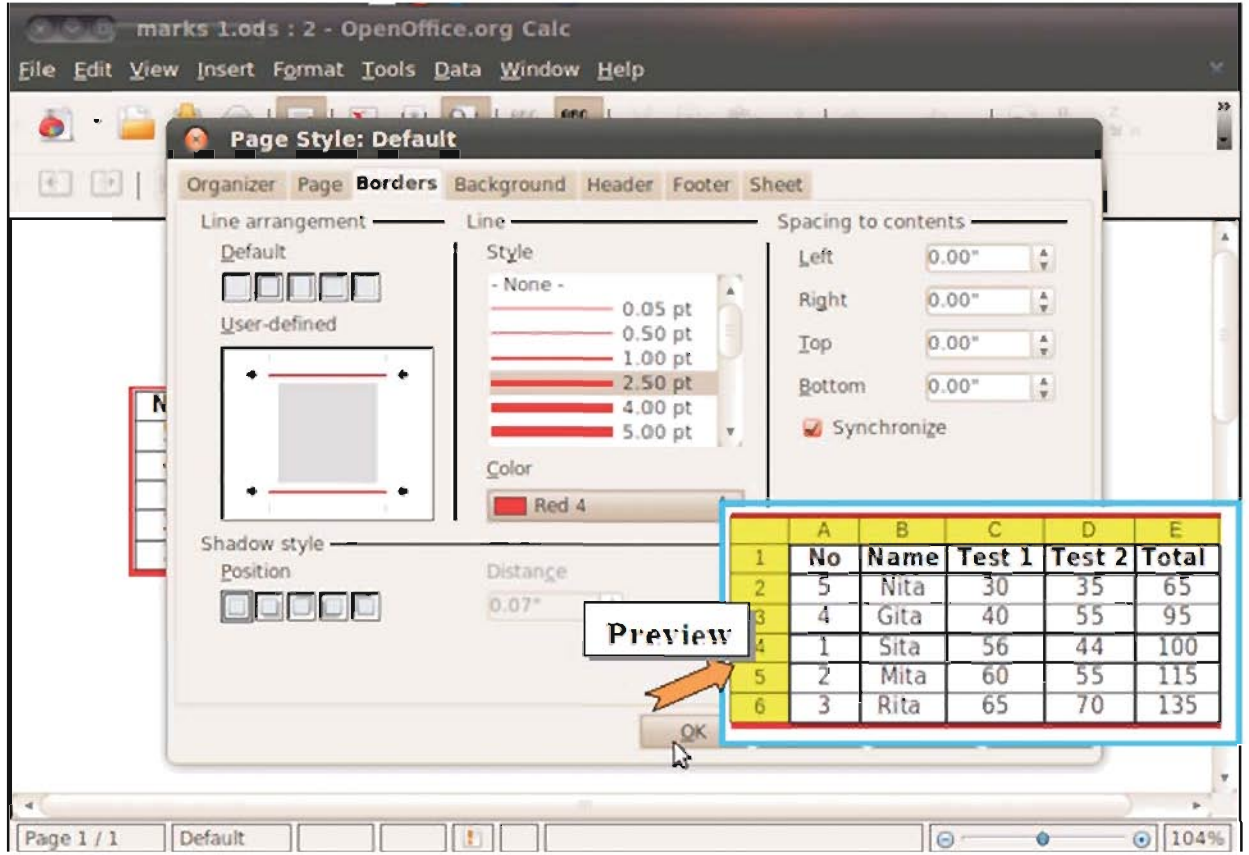
જો તમે આડી હરોળનાં મથાળાં, ઊભી હરોળનાં મથાળાં સાથે સેલ ગ્રીડ (સેલની આસપાસ કિનારીઓ સાથે) છાપવા ઇચ્છતા હોય તો નીચે જણાવ્યા પ્રમાણેની ક્રિયાઓ કરો :

- પસંદ કરો : **Format → Page** એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થશે.
- આના વિકલ્પરૂપે, પસંદ કરો **File → Page Preview** અને **Format Page** બટન દબાવો.
- હવે **Sheet** ટેબ ખોલો (આકૃતિમાં છેલ્લો ટેબ)
- આકૃતિ 6.30માં દર્શાવ્યા પ્રમાણે **Grid** ચેકબોક્સને ટિક કરો (ક્લિક કરવાથી ખરાની નિશાની ચેકબોક્સમાં દેખાશે).
- હવે **Column and row headers** ચેકબોક્સ ટિક કરો.
- ડાયલોગ બોક્સ બંધ કરો.



આકૃતિ 6.30 : column and row headers તેમ જ cell gridsને સેટ કરવી

Borders ટેબનો ઉપયોગ કરીને તમે રંગીન કિનારીઓ પણ છાપી શકો છો. તમે કિનારીઓ (borders) સેટ કરવાના અન્ય વિકલ્પો સાથે કિનારીનો પ્રકાર, કિનારીનો રંગ અને લીટીની જાડાઈ પણ પસંદ કરી શકો છો. આકૃતિ 6.31માં સેલની કિનારી સેટ કરવાના વિકલ્પો દર્શાવ્યા છે. એક વાર તમે પ્રાયલ (પેરામીટર) સેટ કરી દો પછી OK બટન દબાવો. તે પછી તમે પાનાનું પૂર્વાવલોકન કરી શકો. આકૃતિ 6.31માં લાલ રંગની કિનારી તેમજ દાખલ કરેલા ડેટા પીળી પશ્ચાદ્ભૂમિ સાથે ડેટાનું પૂર્વાવલોકન (ભૂરા લંબચોરસને કાળા તીર વડે દર્શાવ્યો છે) દર્શાવેલું છે. જો કે પૂર્વાવલોકન જોવા માટે **Borders**, **cell headers** અને **background** સેટ કર્યા પછી તમારે **Page Preview**માં જવું પડે છે.



આકૃતિ 6.31 : Borders અને Background સેટ કરવાં

જ્યારે દરેક વિકલ્પ અને ફોર્મેટ સેટ થઈ જાય તે પછી તમે સૌથી ઉપરની લીટીમાં રહેલા પ્રિન્ટર આઈકોન ઉપર ક્લિક કરીને દસ્તાવેજ છાપી શકો છો.

પેઈજ બ્રેક બદલવું (Modifying page breaks)

પાનું પૂરું થાય તેની નિશાની (પેઈજ બ્રેક) જોવા અને તેને બદલવા માટે નીચેની ક્રિયાઓ કરો :

- પસંદ કરો : **View → Page Break Preview**
- જો તમે પાનાના દેખાવથી સંતુષ્ટ હો તો તમે તે પ્રિન્ટ કરી શકો છો નહીંતર તમારી જરૂરિયાત પ્રમાણે લીટીને ખેંચીને (ડ્રોગ કરીને) પેઈજ બ્રેક બદલો.

મેનૂનો આદેશ **View → Normal** આપવાથી તમે પાનાનો સામાન્ય દેખાવ જોઈ શકો છો.

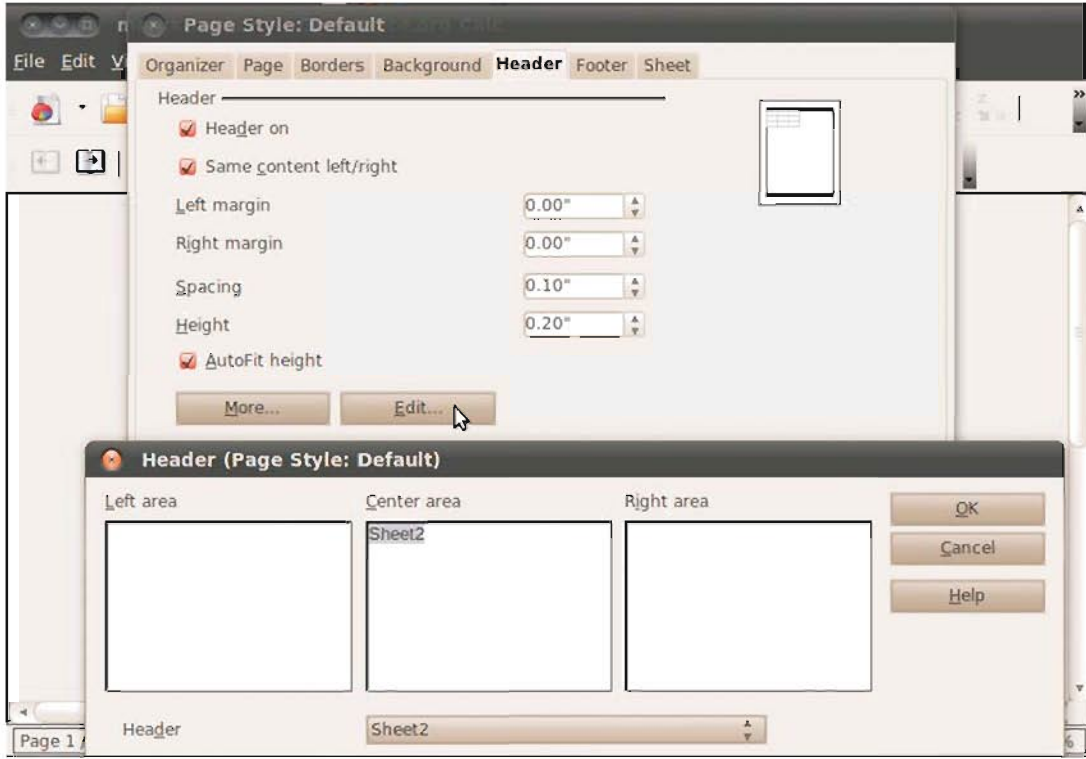
હેડર અને ફૂટર (Header and Footer)

દરેક પાનાની ઉપરની બાજુએ અને દરેક પાનાની નીચેની બાજુએ થોડો વિસ્તાર અનામત રાખવામાં આવે છે. આ ઉપરના અને નીચેના વિસ્તારને અનુક્રમે હેડર (header) અને ફૂટર (footer) કહેવામાં આવે છે.

હેડર બનાવવા માટે નીચે પ્રમાણે આદેશ આપો :

- પસંદ કરો : **File → Page Preview → Format**
- હવે Header ટેબ ખોલો અને તેમાં રહેલાં Edit બટનને દબાવો અને જરૂરી પરિવર્તનો કરો.

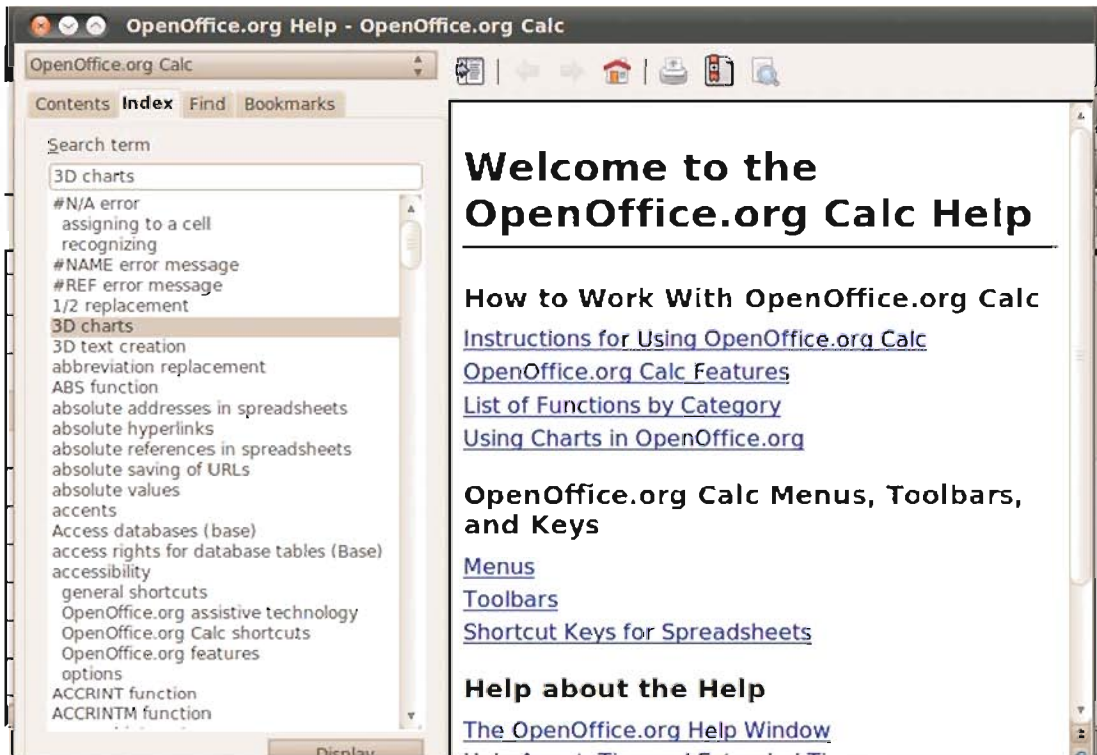
કેલ્સીનાં હેડર અને ફૂટર વિસ્તારને ત્રણ ભાગમાં વિભાજિત કરવામાં આવે છે : ડાબી બાજુનો વિસ્તાર, મધ્ય વિસ્તાર અને જમણી બાજુનો વિસ્તાર. તમે આ વિસ્તારમાં જે કંઈ લખો છો તે આપોઆપ દસ્તાવેજમાં ઉમેરાઈ જશે. આકૃતિ 6.32માં હેડર ઉમેરવાનું દર્શાવેલું છે.



આકૃતિ 6.32 : હેડર ઉમેરવું

મદદ મેળવવી – (Getting help)

કેલ્સીની અંદર જ વિવિધ કાર્યો અને શબ્દોને લગતી મદદ મેળવવા માટેની માર્ગદર્શિકા (manual) સમાયેલી હોય છે. કોઈ ક્રિયા બાબત માહિતી શોધવા માટે તેમાં (કેલ્સીમાં) સર્ચ (search) ક્રિયા પૂરી પાડવામાં આવે છે. આ માટે તમારે ફક્ત **Help → OpenOffice.org Help** આદેશ આપવો પડે છે. જ્યારે તમે મદદ મેળવવા માટેનો આદેશ આપો છો ત્યારે આકૃતિ 6.33માં દર્શાવ્યા પ્રમાણેનો સ્ક્રીન દેખાશે. આનો બીજો સરળ વિકલ્પ કી બોર્ડ ઉપરથી **F1** ફંક્શન કી દબાવવાનો છે.



આકૃતિ 6.33 : મદદ મેળવવી

સારાંશ

આ પ્રકરણમાં આપણે કેલ્સી સ્પ્રેડશીટમાં ડેટાને આકર્ષક રીતે રજૂ કરવા માટે ફોર્મેટિંગ, સોર્ટિંગ (ગોઠવણી) અને વેલિડેટ (ચકાસણી) ક્રિયાઓ બાબત શીખ્યા. આપણે સ્પ્રેડશીટ કક્ષાની ક્રિયાઓ વિશે જાણ્યું કે જે નવા કે હયાત દસ્તાવેજ ખોલવાનું, દસ્તાવેજને કેલ્સીમાં તેમજ અન્ય ફોર્મેટમાં સંગ્રહ કરવાનું તેમજ ફાઈલ ઉપરની અન્ય ક્રિયાઓ જેવી કે દસ્તાવેજને શોધવાનું કે બંધ કરવાનું કાર્ય સરળ બનાવે છે. આ ઉપરાંત શીટ કક્ષાની ક્રિયાઓ જેવી કે શીટનું નામ બદલવું અને શીટનો રંગ બદલવો, શીટ ઉમેરવી કે દૂર કરવી વગેરે બાબત વિશે પણ શીખ્યા. આ પ્રકરણમાં આપણે આ ઉપરાંત સેલ કક્ષાની અને રો તેમજ કોલમ કક્ષાની ક્રિયાઓ બાબત પણ શીખ્યા. આપણે ડેટાની ગોઠવણી, ડેટાની ચકાસણી અને અન્ય વિવિધ કાર્યો માટેની ક્રિયાઓ જેવી કે આપોઆપ સેલમાં માહિતી ભરવાની પ્રક્રિયા (ઓટોફિલ ઓપરેશન), હેડર અને ફૂટર સેટ કરવા, દસ્તાવેજ પ્રિન્ટ કરવું, જોડણીની ચકાસણી કરવી તેમજ મદદ મેળવવી વગેરે બાબત વિશે પણ શીખ્યા.

સ્વાધ્યાય

1. કેલ્સીમાં નવો દસ્તાવેજ કઈ રીતે બનાવી શકાય તે સમજાવો.
2. કેલ્સી વર્કશીટનું નામ કઈ રીતે બદલી શકાય અને તેનો રંગ કઈ રીતે બદલી શકાય તે સમજાવો.
3. કેલ્સીમાં વર્કશીટ કઈ રીતે ઉમેરી શકાય અને વધારાની વર્કશીટ દૂર કરી શકાય તે સમજાવો.
4. સેલમાં રહેલી માહિતી દૂર કરવા ઉપર ટૂંક નોંધ લખો.
5. કેલ્સીમાં ઓટોફિલ ટૂલ કઈ રીતે કાર્ય કરે છે તે સમજાવો.
6. કેલ્સીના સંદર્ભમાં સાપેક્ષ સ્થાનાંક અને નિરપેક્ષ સ્થાનાંક સમજાવો.
7. ડેટા નિસ્ચંદન (ફિલ્ટરિંગ) ઉપર ટૂંક નોંધ લખો.
8. ડેટાની ક્રમબદ્ધ ગોઠવણી (સોર્ટિંગ) ઉપર ટૂંક નોંધ લખો.
9. ડેટા ચકાસણી શું છે ? તે શા માટે જરૂરી છે અને કેલ્સી તે કઈ રીતે કરે છે તે સમજાવો.
10. કેલ્સીમાં પ્રિન્ટ કરતી વખતે હેડર અને ફૂટર કઈ રીતે ઉમેરી શકાય તે સમજાવો.
11. નીચે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :
 - (1) નીચેનામાંથી કઈ કાર્યરીતિ ડેટાને ચોક્કસ ક્રમમાં જેમ કે ચઢતા ક્રમમાં કે ઊતરતા ક્રમમાં ગોઠવવાનું કાર્ય કરે છે ?

(a) ડેટા ફોર્મેટિંગ	(b) ડેટા વેલિડેશન્સ
(c) ડેટા ફિલ્ટરિંગ	(d) ડેટા સોર્ટિંગ
 - (2) સેલમાં ફક્ત તારીખ જ દાખલ કરવા માટે નીચેનામાંથી કઈ કાર્યરીતિ વપરાય છે ?

(a) ડેટા ફોર્મેટિંગ	(b) ડેટા વેલિડેશન્સ
(c) ડેટા ફિલ્ટરિંગ	(d) ડેટા સોર્ટિંગ
 - (3) નીચેનામાંથી કયો વિકલ્પ જ્યારે પસંદ કરવામાં આવે છે ત્યારે બધા ડેટા વેલિડેશન્સ દૂર કરે છે ?

(a) Delete formatting	(b) Delete all
(c) Delete formula	(d) Delete me

- (4) કેલ્સીની કઈ સગવડથી આપણે એક કરતાં વધારે જગ્યાએથી કોઈ શબ્દ બદલી શકીએ છીએ ?
- (a) Find & Replace (b) ફક્ત Replace વડે
- (c) Copy આદેશથી (d) Priview આદેશથી
- (5) પેઈજ પ્રિવ્યૂ નીચેનામાંથી કયા સેટિંગ માટે પરવાનગી આપે છે ?
- (a) Borders (b) Margins
- (c) Column and row heading (d) આપેલા તમામ વિકલ્પો
- (6) જોડણી ચકાસણી માટે કેલ્સીમાં કઈ ફંક્શન કીનો ઉપયોગ થાય છે ?
- (a) F1 (b) F2 (c) F4 (d) F7
- (7) કેલ્સીમાં મદદ મેળવવા માટે નીચેનામાંથી કઈ ફંક્શન કીનો ઉપયોગ થાય છે ?
- (a) F1 (b) F2 (c) F4 (d) F7
- (8) ડેટાને ચોક્કસ ક્રમમાં ગોઠવવા માટેની પ્રક્રિયાને શું કહે છે ?
- (a) સોર્ટિંગ (b) સર્ચિંગ (c) ફિલ્ટરિંગ (d) વેલિડેટિંગ
- (9) બિનજરૂરી ડેટાને ગાળીને દૂર કરવાની પ્રક્રિયાને શું કહે છે ?
- (a) સોર્ટિંગ (b) સર્ચિંગ (c) ફિલ્ટરિંગ (d) વેલિડેટિંગ
- (10) કેવળ માન્ય ડેટાને જ દાખલ કરવાની પ્રક્રિયાને શું કહે છે ?
- (a) સોર્ટિંગ (b) સર્ચિંગ (c) ફિલ્ટરિંગ (d) વેલિડેટિંગ

પ્રાયોગિક સ્વાધ્યાય

1. કેલ્સી દસ્તાવેજને Save as વિકલ્પ વાપરીને અન્ય ફોર્મેટમાં સંગ્રહ કરો. તમે જે ફોર્મેટમાં સંગ્રહ કર્યો હોય તે સોફ્ટવેરનો ઉપયોગ કરી તેને ખોલો.
2. કેલ્સી દસ્તાવેજ ખોલો અને Sheet ટેબ વડે તેમાં 7 શીટ ઉમેરો, ત્યાર પછી તેને મેધધનુષ્યના રંગો જેમ કે જાંબલી, વાદળી રંગ ... વગેરેથી દરેક શીટનો રંગ બદલો. તે પછી ખાતરી કરો કે દરેક શીટ તે પ્રમાણેના રંગની છે કે કેમ એટલે કે વાદળી રંગની શીટ વાદળી રંગથી બદલાઈ છે વગેરે.
3. ઋષા સંખ્યાઓ ઉપર ઓટોફિલ ટૂલનો ઉપયોગ કરો.
4. બે તારીખ સાથે ઓટોફિલ ટૂલ વાપરી જુઓ. એક સેલમાં 15-08-2013 લખો તેની બાજુના સેલમાં તે પછીની તારીખ લખો. આ માહિતીને તે પછીના દસ સેલમાં ડ્રેગ કરો. આ કાર્યરીતિનો ઉપયોગ કરી માસિક કેલેન્ડર બનાવવા પ્રયત્ન કરો.
5. શબ્દકોશમાં તમારું નામ ઉમેરો જેથી તમારા નામ નીચે લાલ લીટી પ્રદર્શિત ન થાય.
6. તમારા મિત્રોની એક યાદી તેમની જન્મતારીખ સાથેની બનાવો. ત્યાર પછી તે યાદીને City અને પછી Namesની ક્રમમાં ગોઠવો. તેનાં શીર્ષકનાં મથાળાં નીચે પ્રમાણે લો :

Name	Address in one line	City	Mobile no	Email	Date of Birth	Month of Birth	Year of Birth
Khushi	1, Gokul Park	Anand	01	July	1995
....							

7. આ સ્વાધ્યાયના ઉદાહરણ 6માં તૈયાર કરેલી મિત્રોની યાદીને એ રીતે ફિલ્ટર કરો કે જેથી ફક્ત એ મિત્રોની માહિતી પ્રદર્શિત થાય જેની જન્મતારીખ ઓગસ્ટ માસમાં હોય.
8. Sorting અને Filtering વિશે Help વડે માહિતી મેળવો. આ વિષયો ઉપર Helpનો ઉપયોગ કરીને નોંધ તૈયાર કરો.
9. ડેટા ચકાસણી(વેલિડેશન્સ)ની નોંધનો ઉપયોગ અને આ સ્વાધ્યાયના ઉદાહરણ 6માં દાખલ કરેલા ડેટાનો ઉપયોગ કરીને date of birth કોલમને ચકાસો જેથી કોઈ વ્યક્તિ તારીખમાં 33 જેવી સંખ્યા દાખલ કરી ન શકે. તમારે ડેટાને એ રીતે ચકાસવા પડે જેથી તે તારીખની કિંમતમાં ફક્ત 1 થી 31ની સંખ્યા જ સ્વીકારે.





કેલ્સીમાં વિધેય

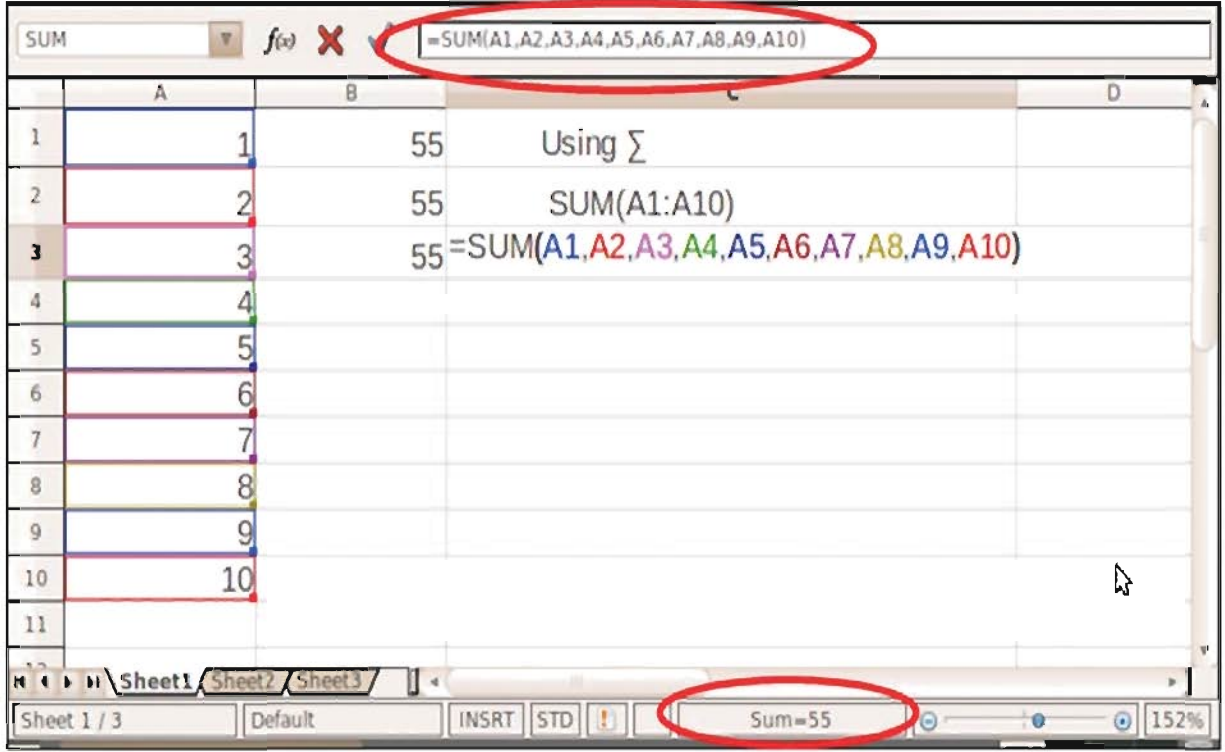
જટિલ નિર્ણયો લેવાની કામગીરી અને પુષ્કળ પ્રમાણમાં ડેટાનું વિશ્લેષણ ધરાવતાં વાસ્તવિક દુનિયાના વિનિયોગોમાં માત્ર ડેટા અને સરળ ગાણિતિક પદાવલિઓ દાખલ કરવી ઘણી વખત પૂરતી નથી. આ પ્રકારનાં કાર્યો કરવા માટે કેલ્સી જેવાં સ્પ્રેડશીટ પ્રોગ્રામ્સ ગાણિતિક, તાર્કિક, આંકડાકીય, સમય અને તારીખ, નાણાંકીય અને શાબ્દિક લખાણ તેમજ સંખ્યાની ગણતરીઓને લગતાં પ્રમાણિત વિધેયોને તેમની અંદર જ સમાવિષ્ટ હોય તે સ્વરૂપે પૂરાં પાડે છે.

કોઈ પદાવલિની જેમ વિધેયની શરૂઆતમાં બરાબરનું ચિહ્ન (=), તે પછી વિધેયનું નામ અને તેના કૌંસમાં એક અથવા વધારે વિધેયનાં ચલ (argument) જણાવવામાં આવે છે. ઘણીવાર વિધેય અને સૂત્રો સાથે "+" (વત્તાનું ચિહ્ન) જેવા પ્રક્રિયક (operator) પણ વાપરી શકાય છે. જો કે વિધેયને એક સાદા શાબ્દિક લખાણ તરીકે ટાળવા માટે પૂર્વગ તરીકે '=' (બરાબર)નું ચિહ્ન વાપરવાની એક સામાન્ય પ્રણાલી છે. વિધેયનાં ચલ (આર્ગ્યુમેન્ટ) તરીકે કોઈ કિંમત, સેલનો સ્થાનાંક, શાબ્દિક લખાણ, અચલ કિંમત અથવા એક કે વધારે વિધેયો હોઈ શકે. પ્રકરણ 5માં આપણે ટૂલબાર ઉપરથી (ખરીદીના બિલના ઉદાહરણમાં) sum વિધેયનો ઉપયોગ કર્યો હતો. તે જગ્યાએ આપણે ટૂલબાર ઉપરથી (સિગ્મા ચિહ્ન Σ વડે દર્શાવેલ છે જે એક પ્રકારનો ટૂંકો રસ્તો છે.) સીધું sum વિધેય લાગુ પાડ્યું હતું. એ જ વિધેય પદ્ધતિસર રીતે **=SUM(A1:A10)** લખી શકાય. જેમાં SUM એ વિધેયનું નામ, સરવાળાની પ્રક્રિયાનો ઉપયોગ કર્યો છે તેનો નિર્દેશ કરે છે, અને A1 તથા A10 કેલ્સીના એ જ વર્કશીટના સેલનો નિર્દેશ કરે છે. સેલ A1 થી A10 એક સાથે (A1:A10) એ એક સેલની વિસ્તાર (range) જણાવે છે જેમાં 10 અલગ અલગ સેલ છે. વિધેયના નામ પછી કૌંસમાં લખેલી માહિતી વિધેય પ્રદેશના ચલ (argument of the function) તરીકે ઓળખાય છે. વિધેયના ચલ તરીકે કોઈ સેલ સ્થાનાંક અથવા સેલ વિસ્તારને બદલે આપણે કોઈ અચલ કિંમત, જેમ કે, **=SUM(23,25)** પણ લખી શકીએ.

એ જ રીતે કદાચ તમે સેલ સ્થાનાંક A1 અને A10 વચ્ચે વિસર્ગ (:) ને બદલે અલ્પવિરામ (,) અથવા અર્ધવિરામ (;)નો ઉપયોગ કરીને વિધેય **=SUM(A1:A10)** પ્રયત્ન કરવા ઇચ્છો પણ ખરા. તમે નિરીક્ષણ કરી શકો કે ',' અથવા ';' નો ફક્ત ઉપયોગ કરવાથી સેલ A1 અને સેલ A10ની કિંમતોનો સરવાળો જ કરી શકાય. પણ જો સેલ A1થી A10 સુધીના બધા જ સેલની કિંમતોનો સરવાળો કરવો હોય તો આપણે ':' અથવા બધા જ સેલ સ્થાનાંક જુદા જુદા એટલે કે **=SUM(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10)** વાપરવા જોઈએ.

આકૃતિ 7.1માં સરવાળાની ક્રિયા ત્રણ રીતે કરી શકાય તે દર્શાવ્યું છે. તમે પસંદ કરેલા સેલમાં કિંમત (SUM=55) અને ફોર્મ્યુલા બારમાં સરવાળાની આખી પદાવલિ જોઈ શકો છો. તમે એ પણ નોંધ કરી હશે કે જ્યારે સૂત્રમાં A1નો નિર્દેશ કરવામાં આવે છે તો તેનો રંગ ભૂરો છે અને સેલ A1નો રંગ પણ ભૂરો છે. આ પદ્ધતિ દાખલ કરેલાં સૂત્રોનો તાળો મેળવવામાં (ફોંસચેક કરવામાં) ખરેખર ખૂબ ઉપયોગી થાય છે.

વિધેય(ફંક્શન)ને સામેલ કરવાની પ્રક્રિયા બધા વિધેયો માટે એકસરખી રહે છે. વિધેયના ચોક્કસ ચલ (આર્ગ્યુમેન્ટ) અને વિધેયના નામની સાચી જોડણી હોવી અનિવાર્ય છે. એટલે કે કોઈ પણ વિધેયનો ઉપયોગ કરવા માટે આપણે તે વિધેયના જરૂરી ચલ સાથે વિધેયનું ભૂલ વગરનું નામ (સાચો સ્પેલિંગ) દાખલ કરવું જરૂરી છે. આને વિધેયની વાક્યરચના (વિધેયના નિયમો syntax) કહેવામાં આવે છે. જો આપણે વિધેયના નામની સાચી જોડણી અને તેના શક્ય ચલ જાણતાં ન હોઈએ તો તે ભૂલમાં પરિણમશે.



આકૃતિ 7.1 : ત્રણ રીતે સરવાળો કરવાની ક્રિયા

સેલમાં વિધેય દાખલ કરવા માટે નીચે પ્રમાણે કાર્ય કરો :

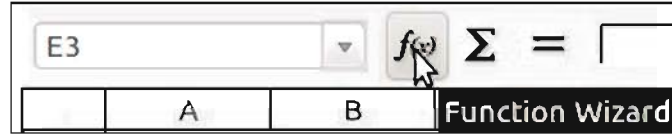
- જે સેલમાં તમારે પરિણામ પ્રદર્શિત કરવું હોય તે સેલ પસંદ કરો.
- હવે વિધેય દાખલ કરવાની શરૂઆત '=' ચિહ્ન સાથે કરો, જો તમે '=' ચિહ્ન ન લખો (અથવા અન્ય માન્ય પ્રચાલક જેમ કે "+"), તો તે વિધેય એક શાબ્દિક લખાણ તરીકે ગણાશે.
- વિધેયનું નામ દાખલ કરો.
- તે પછી વિધેયનાં ચલ દાખલ કરો.
- હવે એન્ટર કી દબાવો.

ફંક્શન વિઝાર્ડ (Function Wizard)

ટૂલબાર ઉપર અનેક વિધેયોનાં આઈકોન રાખવા (સામાન્ય રીતે સ્ક્રીનની સૌથી ઉપરની લીટી) અથવા આપણે તમામ વિધેયોને વાક્યરચના સાથે યાદ રાખવા શક્ય નથી, આથી ઘણાં ખરાં વિધેયો ફંક્શન વિઝાર્ડ (function wizard) વડે દાખલ કરવામાં આવે છે.

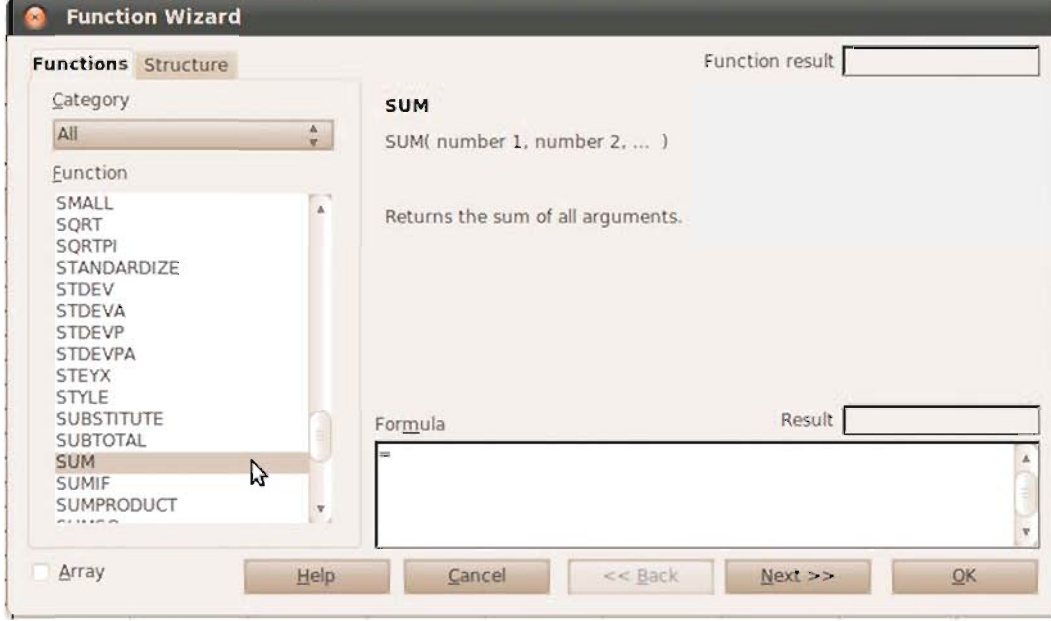
ફંક્શન વિઝાર્ડનો ઉપયોગ નીચે દર્શાવેલી રીતે થાય છે :

- જે સેલમાં પરિણામ પ્રદર્શિત કરવું હોય તે સેલ પસંદ કરો.
- પસંદ કરો : **Insert → Function**. આ કાર્ય તમે ખોલેલા સ્પ્રેડશીટ દસ્તાવેજમાં **Ctrl** અને **F2** દબાવીને પણ કરી શકો છો.
- આના વિકલ્પરૂપે તમે આકૃતિ 7.2માં દર્શાવ્યા પ્રમાણે ફોર્મ્યુલા ટૂલબાર ઉપરનાં **Function Wizard** આઈકોન દબાવીને પણ તે કાર્ય કરી શકો છો.



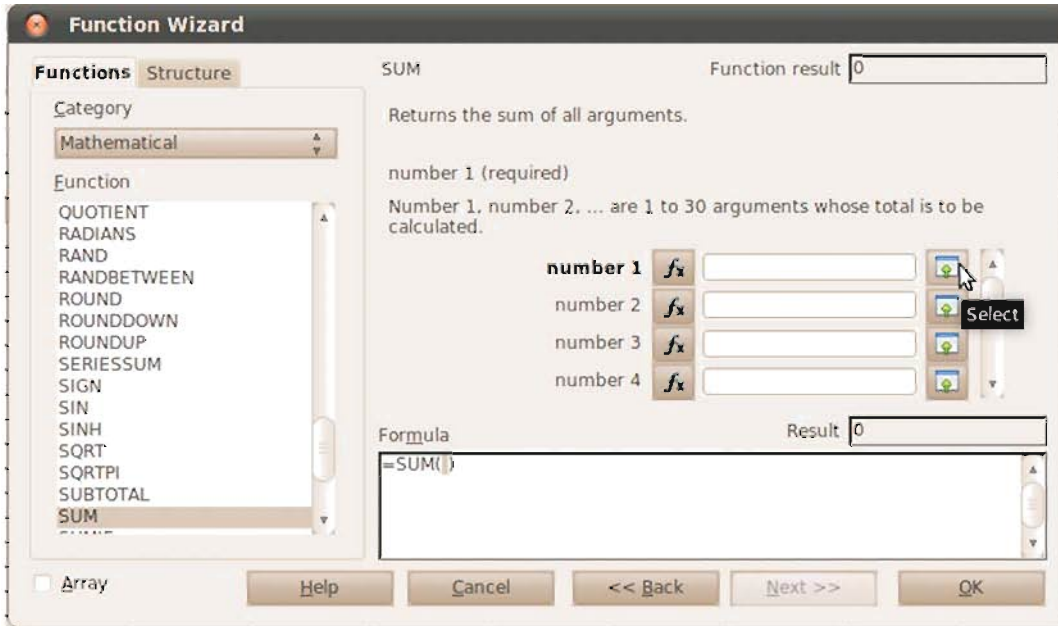
આકૃતિ 7.2 : ફંક્શન વિઝાર્ડ

- આથી એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર દેખાશે. ડાયલોગ બોક્સમાંથી જરૂરી વિધેય પસંદ કરો. આકૃતિ 7.3માં એક લાક્ષણિક ડાયલોગ બોક્સ દર્શાવેલું છે.



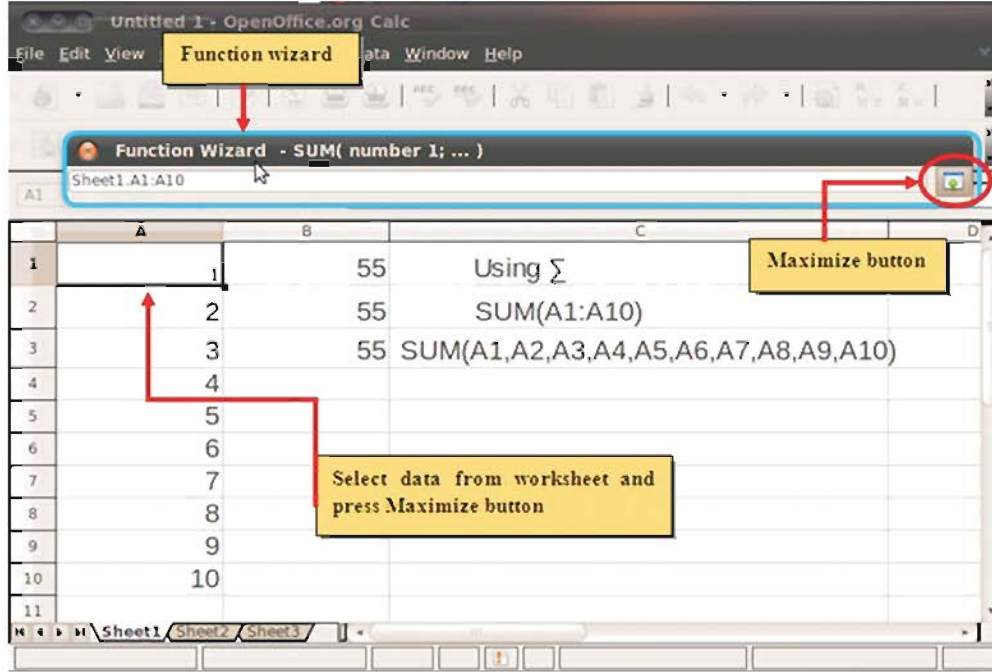
આકૃતિ 7.3 : ફંક્શન વિઝાર્ડનું ડાયલોગ બોક્સ

- SUM** વિધેય પસંદ કરો. તે પછી **Next** બટન દબાવો.
- કિંમતો દાખલ કરો અથવા વર્કશીટમાંથી વિધેયનાં ચલ પસંદ કરો. અહીં તમે 30 ચલની કિંમતો, સ્થાનાંક (નિર્દેશો) અથવા વિસ્તાર એક પછી એક દાખલ કરી શકો છો. આકૃતિ 7.4માં દર્શાવ્યા પ્રમાણે તમે number 1, number 2, ... વગેરે દર્શાવીને ચલની યાદી જોઈ શકો છો.



આકૃતિ 7.4 : વિધેયમાં ચલ દાખલ કરવા

- જો તમે સંખ્યાની કિંમત કે સ્પ્રેડશીટમાં તે ક્યાં છે તે ન જાણતા હો તો તમે વર્કશીટમાંથી સીધા પસંદ કરવાનું ઇચ્છશો.
- Values પસંદ કરવા માટે આકૃતિ 7.4માં માઉસના તીરથી દર્શાવેલા **Select** બટનનો ઉપયોગ કરો.
- જો તમે Select બટન દબાવશો તો તે તમને વર્કશીટના વિસ્તારમાં લઈ જશે. ફંક્શન વિઝાર્ડનું આખું ડાયલોગ બોક્સ એક નાના ટૂલબાર પ્રકારના લંબચોરસમાં પરિવર્તિત થઈ જશે. સ્ક્રીન ઉપર તમે વર્કશીટ વિસ્તાર તેમજ નાનું થયેલું ફંક્શન વિઝાર્ડ જોઈ શકશો. હવે તમે માઉસ વાપરીને સેલ કે સેલ વિસ્તાર પસંદ કરવા માટે મુક્ત છો. આકૃતિ 7.5માં નાનું થયેલું (minimised) ફંક્શન વિઝાર્ડ (ભૂરી રેખા સાથે લંબચોરસ દર્શાવેલ છે) અને વર્કશીટ વિસ્તાર દર્શાવેલો છે. જે ડેટામાં તમે રસ ધરાવતા હો તે વર્તમાન શીટમાં ઉપલબ્ધ ન હોય તો તમે અન્ય શીટ ઉપર પણ જઈ શકો છો. ફંક્શન વિઝાર્ડ ઉપર તે ડેટા ચકાસો. આપણા કિસ્સામાં તે **Sheet1:A1:10** છે.



આકૃતિ 7.5 : Select બટન વડે સેલ વિસ્તાર પસંદ કરવી

- આ કાર્ય કર્યા પછી **Enter** કી અથવા **Maximize** બટન (આકૃતિ 7.5માં દોરેલું વર્તુળ) દબાવો. આકૃતિ 7.5માં દર્શાવ્યા પ્રમાણે ફરી આખું ફંક્શન વિઝાર્ડ સ્ક્રીન ઉપર દેખાશે. તમે જોઈ શકશો કે તમે પસંદ કરેલી ડેટા વિસ્તાર number 1નાં બોક્સમાં દેખવામાં છે.
- કાર્ય પૂર્ણ કરવા માટે અને બંધ કરવા માટે OK બટન દબાવો.

કેલ્સી વિવિધ શ્રેણીઓનાં તેની અંદર જ સમાવેલાં (built-in) અનેક વિધેયો પૂરાં પાડે છે. મુખ્ય શ્રેણીઓનાં વિધેયોમાં ગાણિતિક, આંકડાકીય, તાર્કિક અને શાબ્દિક લખાણનાં ફોર્મેટિંગને લગતાં વિધેયો છે. જ્યારે તમે એક વિધેયના ચલ તરીકે અન્ય કોઈ વિધેયનો ઉપયોગ કરવા ઇચ્છતા હો ત્યારે મૂળ વિધેયના કોંસની અંદર ટેક્સ્ટ કર્સર (text cursor) મૂકીને વિધેયોની યાદીમાંથી નવું વિધેય પસંદ કરો.

ગાણિતિક વિધેયો (Mathematical functions)

જ્યારે આપણે કેલ્સીમાં સંખ્યાઓ સાથે કાર્ય કરીએ છીએ ત્યારે ગાણિતિક વિધેયો આપણને ખૂબ મદદરૂપ બને છે. ચાલો, હવે આપણે કેટલાંક ગાણિતિક વિધેયનો ઉપયોગ જોઈએ:

સંખ્યાની નિરપેક્ષ કિંમત (Absolute values of numbers)

ABS વિધેય ઋણ સંખ્યામાંથી ઋણ ચિહ્ન (–) દૂર કરે છે અને સંખ્યાને ધન બનાવે છે. તે 0 (શૂન્ય) કે ધન સંખ્યામાં કોઈ ફેરફાર કરતું નથી. તેની વાક્યરચના (સિન્ટેક્સ) **ABS (n)** છે, જ્યાં n એક સંખ્યા છે. આકૃતિ 7.6માં ABS વિધેયનું કાર્ય દર્શાવેલું છે. તમે ફોર્મ્યૂલા બાર ઉપર વિધેયનું સૂત્ર (અંકાકાર ચિહ્નિત) પણ જોઈ શકો છો.

EXP		f(x) X		=ABS(B1)	
		A		B	
1	Value of n:			-13	
2	Function used:			=ABS(B1)	
3	Value of ABS(n):			13	
4					

આકૃતિ 7.6 : ABS વિધેયનું કાર્ય

જ્યારે તમારે ફક્ત ધન સંખ્યાની જ જરૂર હોય તે સમયે ABS વિધેય ઘણું ઉપયોગી થાય છે. ધારો કે તમે એક પદાવલિની અંદર બીજી પદાવલિ વાપરી રહ્યા છો. આવા સમયે કેલ્સી સૌપ્રથમ અંદરની પદાવલિની કિંમત શોધશે અને પછી અંદરની પદાવલિની કિંમતનો ઉપયોગ બહારની પદાવલિમાં એક ચલ (આર્ગ્યુમેન્ટ) તરીકે કરશે. પણ ધારો કે બહારની પદાવલિ (જેમ કે વર્ગમૂળ) ફક્ત ઋણ ન હોય તેવી કિંમત જ (ધન અથવા 0) માન્ય રાખે છે. આવા સમયે જો અંદરની પદાવલિનું પરિણામ ઋણ આવે તો શું થાય ? અંદરની પદાવલિનું ઋણ પરિણામ બહારની પદાવલિને અમાન્ય બનાવી દે છે. આવી સ્થિતિ ટાળવા માટે આપણે અંદરની પદાવલિના પરિણામની નિરપેક્ષ કિંમતને બહારની પદાવલિમાં મોકલીએ છીએ.

ઘાતાંકીય વિધેય (Exponential function)

EXP વિધેય ઘાતાંકીય વિધેય e^x ની કિંમત પાછી આપે છે. જ્યાં x એ આપેલી સંખ્યા છે અને e ની કિંમત આશરે 2.718281828 બરાબર છે. તેની વાક્યરચના **EXP(Number)** છે. દા.ત. જો આપણે કોઈ સેલમાં **=EXP(1)** લખીએ તો તે e^1 ગણાશે અને તેની કિંમત 2.72 પાછી આપશે. આકૃતિ 7.7માં આ વિધેય સમજાવ્યું છે. તમે ફોર્મ્યુલા બાર (અંડાકાર ચિહ્નિત) ઉપર વિધેયનું સૂત્ર જોઈ શકશો.

EXP		f(x) X		=EXP(B1)	
		A		B	
1	Value of n:			1	
2	Function used:			=EXP(B1)	
3	Value of EXP(n):			2.71828182845904	
4					

આકૃતિ 7.7 : EXP ફંક્શન

ક્રમગુણિત વિધેય (ફેક્ટોરિઅલ ફંક્શન - Factorial function)

FACT વિધેય આપેલી સંખ્યાની ક્રમગુણિત વિધેયની કિંમત પાછી આપે છે (1થી આપેલી પૂર્ણાંક સંખ્યા n સુધીનો ગુણાકાર). આ વિધેયની વાક્યરચના **FACT(n)** છે જ્યાં n એ પૂર્ણાંક સંખ્યા છે.

ઉદાહરણ તરીકે, જો આપણે **=FACT(3)**, લખીએ તો કેલ્સી તેને પદાવલિ $(1 * 2 * 3)$ ગણાશે અને પરત કિંમત 6 કરશે. એ જ રીતે FACT(6) આપણને 1થી 6 સંખ્યાનો ગુણાકાર (એટલે કે $1*2*3*4*5*6$) પરત આપશે, કે જે 720 છે. આકૃતિ 7.8માં સંખ્યા 6 સાથે ક્રમગુણિત વિધેય દર્શાવેલું છે.

FACT		f(x) X ✓	=FACT(B1)
1	Value of n:	6	
2	Function used:	=FACT(B1)	
3	Value of FACT(n):	720	

આકૃતિ 7.8 : FACT વિધેય

પ્રાકૃતિક લઘુગણક (નેચરલ લોગેરિધમ - Natural logarithm)

LN વિધેય આપેલી ધન સંખ્યાની પ્રાકૃતિક લઘુગણક (લઘુગણકનો આધાર e) કિંમત પરત આપે છે. આ વિધેયની વાક્યરચના **LN (Number)** છે, જ્યાં number એ ધન સંખ્યા છે. દા.ત., આપણે કોઈ સેલમાં =LN(8) લખીએ તો, આકૃતિ 7.9માં દર્શાવ્યા પ્રમાણે તે 2.08 (log_e 8) જવાબ તરીકે આપશે.

LN		f(x) X ✓	=LN(B1)
1	Value of n:	8	
2	Function used:	=LN(B1)	
3	Value of LN(n):	2.0794415417	

આકૃતિ 7.9 : LN વિધેય

લઘુગણક આધાર 10 સાથે (Logarithm base 10)

LOG10 વિધેય આપેલી ધન સંખ્યાની લઘુગણક આધાર 10 સાથે કિંમત પરત આપે છે. તેની વાક્યરચના **LOG10(Number)** છે, જ્યાં number એ ધન સંખ્યા છે. ઉદાહરણ તરીકે, LOG10(6)ને LOG₁₀6 ગણવામાં આવે છે અને આકૃતિ 7.10માં દર્શાવ્યા પ્રમાણે 0.77 કિંમત પરત આપે છે.

LOG10		f(x) X ✓	=LOG10(B1)
1	Value of n:	6	
2	Function used:	=LOG10(B1)	
3	Value of LOG10(n):	0.7781512504	

આકૃતિ 7.10 : LOG₁₀ વિધેય

ઘાત (Power)

ઘાત (પાવર) વિધેય સંખ્યાની ઘાતની ગણતરી કર્યા પછીની કિંમત પરત આપે છે. તેની વાક્યરચના **POWER(Number, Power)** છે. જો કોઈ સેલમાં આપણે =POWER (10,3) લખીએ તો તે 1000 (10^3) કિંમત પરત આપશે. તે 10^3 ના સંકેત બરાબર છે. જુઓ આકૃતિ 7.11.

POWER		f(x) X ✓	=POWER(B1,B2)
	A	B	
1	Value of n:	10	
2	Value of power:	3	
3	Function used:	=POWER(B1,B2)	
4	Value of POWER(n, power):	1000	
5			

આકૃતિ 7.11 : POWER વિધેય

વિધેયના અનેક ચલનો ગુણાકાર (Product of many arguments)

PRODUCT વિધેય વધારેમાં વધારે 30 વિધેય પ્રદેશનાં ચલ(x1, x2, ..., x30)નો ગુણાકાર પાછો આપે છે. વિધેયનો દરેક ચલ કોઈ એક સેલ કે સેલ વિસ્તાર હોઈ શકે. આ વિધેયની વાક્યરચના **PRODUCT (n1, n2, n3, ..., n30)** છે. ધારો કે આપણે આકૃતિ 7.12માં દર્શાવ્યા પ્રમાણે વર્કશીટમાં ડેટા દાખલ કરેલો છે.

A4		f(x) Σ =	=PRODUCT(A1:A2, B1:B2, D1:D2)		
	A	B	C	D	E
1	10	1		3	
2	20	2		1	
3					
4	1200				
5					

આકૃતિ 7.12 : PRODUCT વિધેયનો ઉપયોગ કરીને વિધેયનાં અનેક ચલનો ગુણાકાર

ધારો કે કોઈ સેલમાં આપણે =PRODUCT(A1:A2, B1:B2, D1:D2) લખીએ છીએ. આ વિધેય પરિણામ 1200 આપશે. આથી, આપણે કહી શકીએ કે આ product વિધેય સેલમાં રહેલી બધી કિંમતો, સેલનો વિસ્તાર અથવા એક કરતાં વધારે સેલના વિસ્તારને ધ્યાનમાં લે છે અને બધાનો ગુણાકાર કરીને પરિણામ આપે છે. આકૃતિ 7.12માં ઊભી હરોળ (કોલમ) A, કોલમ B અને કોલમ Dમાં કુલ 6 કિંમતો લખેલી છે. નોંધ કરો કે કોલમ C ખાલી છે.

વર્ગમૂળ (Square root) (SQRT વિધેય)

SQRT વિધેય ધન સંખ્યાનું વર્ગમૂળ પાછું આપે છે. આ વિધેયની વાક્યરચના **SQRT(Number)** છે, જ્યાં Number એ ધન સંખ્યા છે. આપણે 100નું વર્ગમૂળ શોધવા માટે કોઈ એક સેલમાં **=SQRT(100)** ફક્ત લખવું પડે. આકૃતિ 7.3માં દર્શાવ્યા પ્રમાણે તે કિંમત 10 પરત આપશે.

F7	f(x)	Σ	=	
	A	B	C	D
1	Value of n:	76.660000	45.560000	-56.547890
2	Decimal places:	2	0	3
3	Function used:	ROUND(B1,B2)	ROUND(C1,C2)	ROUND(D1,D2)
4	Value of ROUND(n,decimal places):	76.66	46	-56.548

આકૃતિ 7.15 : ROUND વિધેયનાં ઉદાહરણો

ROUND વિધેય જેવાં જ અન્ય બે વિધેયો ROUNDDOWN અને ROUNDUP છે જેની વાક્યરચના ROUND જેવી જ છે. આ વિધેયો ઉપર વિવિધ સંખ્યાઓ લઈ પ્રયોગો કરી જુઓ અને તેમની વચ્ચેનો તફાવત જાણો.

ટ્રન્કેટ (Truncate – સ્થૂળમૂલ્ય) (TRUNC વિધેય)

ટ્રન્કેટ વિધેય(TRUNC)માં બે ચલ હોય છે, જેમાં પહેલો ચલ એ સંખ્યા છે અને બીજો ચલ એ સંખ્યાના અપૂર્ણાંક ભાગમાંથી કેટલા અંક સંખ્યા સાથે કાપવાના છે એ સંખ્યા દર્શાવે છે. ટ્રન્કેટ વિધેય જો જણાવેલ હોય તો દશાંશબિંદુ પછી કેટલાક અંક રાખીને આપેલી સંખ્યાના અપૂર્ણાંક ભાગને કાપી નાખે છે. જો સંખ્યામાં દશાંશબિંદુ પછી y અંક હોય અને TRUNC વિધેયમાં બીજું ચલ n આપેલું હોય તો આપેલી સંખ્યામાંથી આખો જ અપૂર્ણાંક ભાગ કાપી નાખવામાં આવે છે. આ વિધેયની વાક્યરચના **TRUNC(Number,Places)** છે, જ્યાં Number એ આપેલી સંખ્યા છે અને Places એ દશાંશબિંદુ પછી કેટલા અંક રાખવા છે તે જણાવે છે.

નીચે આપેલાં ઉદાહરણોનો અભ્યાસ કરો :

=TRUNC(1.239,2)નું પરિણામ 1.23 મળે છે અને અંક 9 ગુમાવીએ છીએ.

=TRUNC(12.5)નું પરિણામ 12 મળે છે અને અંક 5 ગુમાવીએ છીએ.

=TRUNC(-15.72)નું પરિણામ -15 મળે છે અને અંક 7 અને 2 ગુમાવીએ છીએ.

ઉપરનાં ઉદાહરણમાં TRUNC વિધેયની જગ્યાએ INT વિધેય વાપરી જુઓ અને પરિણામના તફાવત વિશે વિચાર કરો. તમે જોઈ શકશો કે TRUNC વિધેય આપેલી સંખ્યામાંથી નિર્દેશિત ભાગ ફક્ત કાપી જ નાખે છે. તે સંખ્યાને રાઉન્ડ કરશે નહીં. આ ઉદાહરણો આકૃતિ 7.16માં દર્શાવ્યા છે.

E7	f(x)	Σ	=	
	A	B	C	D
1	Value of n:	1.239	12.5	-15.72
2	Decimal places:	2	--	--
3	Function used:	TRUNC(B1,B2)	TRUNC(C1)	TRUNC(D1)
4	Value of TRUNC(n,decimal places):	1.23	12.00	-15.00

આકૃતિ 7.16 : TRUNC વિધેયનાં ઉદાહરણો

ROUND વિધેયમાં આપેલાં ઉદાહરણોનો પણ અભ્યાસ કરો અને એ જ ઉદાહરણોનો TRUNC વિધેય સાથે પ્રયોગ કરી જુઓ.

આંકડાકીય વિધેયો (Statistical functions)

કેલ્સી લાક્ષણિક ગાણિતિક વિધેયો ઉપરાંત કિંમતોની શ્રેણી ઉપર અનેક વિવિધ આંકડાકીય વિધેયો પણ તમને પૂરા પાડે છે. આ આંકડાકીય વિધેયો ડેટાનો સંચય કરવામાં, તેનું વિશ્લેષણ કરવામાં તેમજ તેને સમજવામાં તથા રજૂઆત કરવામાં મદદરૂપ થાય છે જેથી આગાહી કે ભવિષ્યવાણી કરવાનું અને નિર્ણયો લેવાનું કાર્ય સરળ બની રહે છે. ચાલો, હવે આપણે કેટલાક પ્રચલિત આંકડાકીય વિધેયોનો અભ્યાસ કરીએ.

સરેરાશ (Average) – AVERAGE વિધેય

AVERAGE વિધેય આપેલી સંખ્યાઓની સરેરાશ કિંમત પાછી આપે છે. AVERAGE એ અંકગણિતના મધ્યક તરીકે પણ ગણવામાં આવે છે. તમારે યાદીમાં રહેલી બધી સંખ્યાઓનો સરવાળો કરી તેને યાદીમાં રહેલી કુલ સંખ્યા (યાદીનું કદ) વડે ભાગાકાર કરવો પડે. તમે આવી મહત્તમ 30 કિંમતો લઈ શકો જેને અર્ધવિરામ(;)થી જુદી પાડવામાં આવે છે. હવે તમે જાણો છો કે કિંમતોને બદલે તમે સેલ સ્થાનાંક અથવા માન્ય સેલનો વિસ્તાર પણ દાખલ કરી શકો. આ વિધેયની વાક્યરચના **AVERAGE(x1,x2,...,x30)** છે. ઉદાહરણ તરીકે, જો તમે કોઈ સેલમાં **=AVERAGE(10,7,6,8,9,5)** લખો તો તમને જવાબ 7.5 મળશે. આકૃતિ 7.17માં AVERAGEનું કાર્ય દર્શાવેલું છે.

B3		$f(x)$	Σ	=	=AVERAGE(B1:G1)		
	A	B	C	D	E	F	G
1	Numbers:	10	7	6	8	9	5
2	Function used:	AVERAGE(B1:G1)					
3	Average of Numbers:	7.5					

આકૃતિ 7.17 : AVERAGE વિધેયનું ઉદાહરણ સાથે નિદર્શન

મધ્યક (Mean) – GEOMEAN વિધેય

GEOMEAN વિધેય આપેલી સંખ્યાઓની સમગુણોત્તર મધ્યક (geometric mean) કિંમત પાછી આપે છે. ઋણ ન હોય તેવી n સંખ્યાઓનો સમગુણોત્તર મધ્યક મેળવવા માટે બધી જ n સંખ્યાઓનો ગુણાકાર કરી પછી તેનું n^{th} મૂળ કાઢવામાં આવે છે. તમે મહત્તમ 30 ચલ (કિંમતો, સ્થાનાંક અથવા વિસ્તાર) લઈ શકો છો. આ વિધેયની વાક્યરચના **GEOMEAN(x1, x2 ,..., x30)** છે. જો તમે સ્પ્રેડશીટના કોઈ સેલમાં **=GEOMEAN(10,7,6,8,6,5)** લખો તો 6.8 જવાબ મળશે.

કેલ્સીમાં અન્ય પ્રકારનું મધ્યક પણ ઉપલબ્ધ છે જે HARMEAN (હરાત્મક મધ્યક harmonic mean) છે. હરાત્મક મધ્યકની વાક્યરચના **HARMEAN(x1,x2,...,x30)** છે. ઉપર જણાવેલા ડેટાનું હરાત્મક મધ્યક ઉપરનું વિધેય વાપરીને શોધો.

મધ્યસ્થ (Median) – MEDIAN વિધેય

મધ્યસ્થ એ આપેલી શ્રેણીને ક્રમમાં ગોઠવ્યા પછીની મધ્ય સંખ્યા છે. જો તમે એક નિશ્ચિત મધ્ય સંખ્યા n મેળવી શકો (શ્રેણીમાં કુલ બેકી સંખ્યામાં ચલ હોય ત્યારે) તો વચ્ચેની બે સંખ્યાની સરેરાશ લો. MEDIAN વિધેય મહત્તમ 30 ચલનું મધ્યસ્થ શોધી શકે છે. વિધેયની વાક્યરચના **MEDIAN(x1,x2,...,x30)** છે. જો સ્પ્રેડશીટના કોઈ સેલમાં તમે **=MEDIAN(10,7,6,8,6,5)** લખો તો 6.5 જવાબ મળશે.

બહુલક (Mode) – MODE વિધેય

MODE વિધેય આપેલા મહત્તમ 30 ચલ કે જે ફક્ત એક કિંમત અથવા સેલનો વિસ્તાર હોઈ શકે, તે ડેટા સેટની સૌથી વધારે સામાન્ય કિંમત પાછી આપે છે. તેની વાક્યરચના **MODE(x1, x2,...,x30)** છે. દા.ત. જો તમે MODE વિધેયને **=MODE(10,7,6,8,6,5)** લખશો તો તે 6 જવાબ આપશે. આકૃતિ 7.18માં વિવિધ આંકડાકીય વિધેયો દર્શાવેલા છે.

	A	B	C	D	E	F	G
1	Numbers:	10	7	6	8	6	5
2	Function used:	AVERAGE(B1:G1)					
3	Average of Numbers:	7					
4							
5	Function used:	MODE(B1:G1)					
6	Mode of Numbers:	6					
7							
8	Function used:	GEOMEAN(B1:G1)					
9	Geo.Mean of Numbers:	6.822					
10							
11	Function used:	MEDIAN(B1:G1)					
12	Median of Numbers:	6.5					

આકૃતિ 7.18 : GEOMEAN, MEDIAN અને MODE વિધેયો

CountA વિધેય

COUNTA વિધેય કોઈ કિંમતનો સંગ્રહ કુલ કેટલા સેલમાં અથવા ચલમાં કરવામાં આવ્યો છે તેની સંખ્યા આપે છે. તમે મહત્તમ 30 ચલ (કિંમતો, સ્થાનાંક અથવા વિસ્તાર) અર્ધવિરામનાં ચિહ્ન(;)થી અલગ પાડીને આપી શકો છો. તેમાંથી કેટલાક ચલમાં અમુક કિંમત (કોઈ પણ પ્રકારની) હોઈ શકે અને કેટલાક ખાલી પણ હોઈ શકે. COUNTA વિધેય અમુક કિંમત ધરાવતાં સેલની કુલ સંખ્યા આપે છે. ખાલી સેલની ગણતરી તેમાં થશે નહીં. આ વિધેયની વાક્યરચના **COUNTA(x1, x2, ... , x30)** છે. જો તમે સેલમાં **=COUNTA(10,7,6,8,6,5)** લખશો તો જવાબ 6 મળશે.

મોટામાં મોટી કિંમત (Largest value) – LARGE વિધેય

LARGE વિધેય આપેલી x સંખ્યાના ગણમાંથી kth સૌથી મોટી આંકડાકીય કિંમત પાછી આપે છે. તેની વાક્યરચના **=LARGE(x, k)** છે, જ્યાં x એ અમુક કિંમતો ધરાવતી સેલ વિસ્તારનો નિર્દેશ કરે છે અને k એ સંખ્યાનું સ્થાન દર્શાવે છે. આ વિધેય આપેલી સંખ્યાઓને ઊતરતા ક્રમમાં ગણીને k સ્થાનની સંખ્યા પાછી આપે છે. ઉદાહરણ તરીકે, **10, 7, 6, 8, 6 અને 5**ની કિંમતોના ગણમાં LARGE વિધેયમાં 2 કિંમત આપીએ તો આપણે જવાબ 8 મેળવીશું. આ સંખ્યા 8 એ આપેલી સંખ્યાનાં ગણમાં બીજી (2nd) મોટી કિંમત છે.

અહીં વિધેયમાં ચલ kની કિંમત તમે ઇચ્છિત પૂર્ણાંક સંખ્યા આપી શકો સિવાય કે તે શૂન્ય ન હોઈ શકે અથવા આપેલી સંખ્યાના ગણની કુલ સંખ્યા કરતાં વધારે ન હોઈ શકે.

નાનામાં નાની કિંમત (Smallest value) – SMALL વિધેય

SMALL વિધેય આપેલી x સંખ્યાના ગણમાંથી kth નાનામાં નાની આંકડાકીય કિંમત પાછી આપે છે. તેની વાક્યરચના **=SMALL (x, k)** છે, જ્યાં x એ અમુક કિંમતો ધરાવતી સેલ વિસ્તારનો નિર્દેશ કરે છે અને k એ સંખ્યાનું સ્થાન દર્શાવે છે.

આ વિધેય આપેલી સંખ્યાઓને ચઢતા ક્રમમાં ગણીને k સ્થાનની સંખ્યા પાછી આપે છે. ઉદાહરણ તરીકે, **10, 7, 6, 8, 6, 5**ની કિંમતોના સેટમાં SMALL વિધેયમાં 4 કિંમત આપીએ તો જવાબ 7 મેળવીશું કે જે આપેલી સંખ્યાઓનાં સેટમાં ચોથી નાનામાં નાની સંખ્યા છે. આકૃતિ 7.19માં LARGE, SMALL અને COUNTA વિધેય ઉદાહરણો સાથે દર્શાવ્યા છે.

H12	f(x)	Σ	=				
	A	B	C	D	E	F	G
1	Numbers:	10	7	6	8	6	5
2	Function used:	COUNTA(B1:G1)					
3	Count of Numbers:	6					
4							
5	Function used:	LARGE(B1:G1;2)					
6	Second Largest Value:	8					
7							
8	Function used:	SMALL(B1:G1;4)					
9	Fourth Smallest Value:	7					
10							

આકૃતિ 7.19 : LARGE, SMALL અને COUNTA વિધેયોનાં ઉદાહરણો

રેન્ક (Rank) – RANK વિધેય

રેન્ક વિધેય કોઈ આપેલી સંખ્યાઓના ગણમાં કોઈ નક્કી કરેલી સંખ્યાનો ક્રમ પરત આપે છે. આ માટે તમારે કોઈ ક્રિમત (અથવા ક્રિમતનો સ્થાનાંક) કે જેનો ક્રમ તમારે જાણવો છે, સંખ્યાઓનો ગણ (સેલનો વિસ્તાર) અને સંખ્યાઓને ગોઠવવાનો ક્રમ; ચઢતા ક્રમ માટે 1 અને ઊતરતા ક્રમ માટે 0 જણાવવો પડે છે. આ વિધેય આપેલી સંખ્યાઓના સેટને ચઢતા કે ઊતરતા ક્રમમાં ગોઠવશે અને આપણે જણાવેલી સંખ્યાનું સ્થાન (પહેલું, બીજું, ત્રીજું... વગેરે) કેટલામું છે તે ક્રિમત પરત આપે છે.

આ વિધેયની વાક્યરચના **RANK(number, set, order)** છે. ધારો કે તમે પરીક્ષામાં 67 ગુણ મેળવ્યા છે. તમારા પાંચ મિત્રોએ 47, 56, 78, 59 અને 66 ગુણ મેળવ્યા છે. તમે જાણો છો કે તમારા મિત્રોમાં તમારો રેન્ક બીજો છે; સૌથી વધારે ગુણનો પહેલો નંબર એટલે કે ગુણ ઊતરતા ક્રમમાં છે. આ માહિતી તમે ડેટા તરફ ફક્ત નજર કરીને જ મેળવી શકો કારણ કે તમે ફક્ત તમારા મિત્રોની જ (છ મિત્રોની) માહિતી ધ્યાનમાં લીધી છે. પણ વિચારો કે જો ઘણા વિદ્યાર્થીઓની માહિતી હોય તો શું થાય ? જવાબ ઘણો સહેલો છે. આ માટે RANK વિધેય વાપરો. ઉદાહરણ તરીકે, તમારો નંબર જાણવા માટે (દેખીતી રીતે ઉપરથી), તમે RANK વિધેય **=RANK(67, A1:F1, 0)** વાપરી શકો અને જવાબ તમારો નંબર 2 હશે. આકૃતિ 7.20માં RANK વિધેય દર્શાવ્યું છે.

મહત્તમ (Maximum) - MAX વિધેય

MAX વિધેયમાં વધારેમાં વધારે 30 ચલ (એક સેલ અથવા સેલનો વિસ્તાર) આપી શકાય અને આ ચલમાંથી મોટામાં મોટી સંખ્યા (મહત્તમ ક્રિમત) પાછી મળે છે.

આ વિધેયની વાક્યરચના **MAX(x1, x2 ,..., x30)** છે. ઉપર જણાવેલા વિદ્યાર્થીઓના ગુણના ઉદાહરણમાં મહત્તમ ગુણ શોધવા માટે તમે MAX વિધેયનો ઉપયોગ **=MAX(67,47,56,78,59,66)** આપીને કરી શકો છો. આકૃતિ 7.20માં જણાવ્યા પ્રમાણે તે વિધેય 78 જવાબ પરત આપશે.

H12	f(x)	Σ	=				
	A	B	C	D	E	F	G
1	Numbers:	67	47	56	78	59	66
2	Function used:	RANK(67,B1:G1,0)					
3	RANK of 67:	2					
4							
5	Function used:	MAX(B1:G1)					
6	Largest Value:	78					
7							
8	Function used:	MIN(B1:G1)					
9	Smallest Value:	47					
10							

આકૃતિ 7.20 : RANK, MAX અને MIN વિધેયનો ઉપયોગ

લઘુતમ (Minimum) - MIN વિધેય

MIN વિધેયમાં વધારેમાં વધારે 30 ચલ (એક સેલ અથવા સેલ વિસ્તાર) આપી શકાય અને આ ચલમાંથી નાનામાં નાની સંખ્યા (લઘુતમ સંખ્યા) પાછી મળે છે. આ વિધેયની વાક્યરચના **MIN(x1, x2 ,..., x30)** છે. ઉપર જણાવેલા વિદ્યાર્થીઓના ગુણના ઉદાહરણમાં લઘુતમ ગુણ શોધવા માટે તમે MIN વિધેયનો ઉપયોગ **=MIN(67,47,56,78,59,66)** આપીને કરી શકો છો. આકૃતિ 7.20માં દર્શાવ્યા પ્રમાણે તે વિધેય 47 જવાબ પરત આપશે.

નાણાંની ગણતરી (Calculation with money)

ચલણી નાણાંની ગણતરી માટે કેલ્સી નાણાંકીય વિધેયો પૂરાં પાડે છે. ધારો કે તમારા એક કાકા તમારી પાસે આવે છે અને કહે છે :

“બેટા, તારા કમ્પ્યુટરનો ઉપયોગ કરીને મને કહે કે જો કોઈ બેંકમાંથી હું નાણાંનું ઋણ લઉં અને તેને ચૂકવવા માટે દરેક સમયે એકસરખા હપ્તા ચૂકવું અને વ્યાજનો દર એકસરખો રહે તો કેટલાં વર્ષનો સમય લાગશે ?”

આ કાર્ય માટે તમારે ઘણી ગણતરી કરવી પડશે અથવા એક અટપટો કમ્પ્યુટર પ્રોગ્રામ લખવો પડશે. કેલ્સી તમારું કામ સરળ બનાવી દે છે. તમારે ફક્ત સેલમાં એક વિધેય લખવું પડે છે :

=NPER(7.5%,-12000,100000)

અહીં NPERનો ઉપયોગ તમે ઉછીના લીધેલા પૈસાની ચૂકવણી માટેના સમયગાળાની ગણતરી માટે થયો છે. આ વિધેયનું પહેલું ચલ (7.5%) એ અચળ વ્યાજનો દર છે. બીજું ચલ -12000 એ વાર્ષિક હપ્તાની રકમ (રૂપિયામાં) છે (તે ઋણ છે કારણ કે દરેક હપ્તો ચૂકવવામાં આવે ત્યારે કુલ ઉછીની લીધેલી રકમમાંથી આ રકમ બાદ કરવી પડે); અહીં તમારા કાકા 1000 રૂ. પ્રતિ માસ પાછા ચૂકવે છે અને વર્ષમાં કુલ 12000 રૂ. પાછા આપે; ત્રીજું ચલ એ કુલ ઉછીની લીધેલી રકમ છે. જ્યારે તમે આ સૂત્ર કેલ્સીના કોઈ સેલમાં લખો છો ત્યારે તે 13.56 વર્ષ જવાબ પાછો આપશે. આ એ જ જવાબ છે જે તમારા કાકા જાણવા ઇચ્છતા હતા. આ ઉદાહરણ આકૃતિ 7.21માં દર્શાવેલ છે.

E7 f(x) Σ =			
	A	B	C
1	Total amount of loan taken:	100000	(Rs.)
2	Interest rate in %:	7.50%	(Constant rate)
3	Yearly total installment:	12000	(Rs. 1000 per month)
4	Function used: NPER(B2, -B3, B1)		
5	Duration in years :	13.562227330178	(Years)

આકૃતિ 7.21 : NPER વડે ઉછીના લીધેલાં નાણાં પાછા ચૂકવવા માટેનો સમયગાળો

NPER વિધેયની વાક્યરચના **NPER(Interest,Instalment,Loan,Future,Type)** છે, અહીં interest - વ્યાજનો દર, Instalment - હપ્તાની રકમ, Loan - ઉછીની લીધેલી રકમ, Future - છેલ્લો હપ્તો ચૂકવ્યા પછી રહેલી રકમ (સામાન્ય રીતે આપણે ઉછીની લીધેલી રકમની પૂરેપૂરી ચૂકવણી કરીએ છીએ) અને Type હપ્તાના સમયગાળાની પહેલાં ચૂકવણી કરવામાં આવે છે કે નહીં, (જો સમયગાળાની પહેલાં હપ્તો આપવાનો હોય તો તેની કિંમત (typeની કિંમત) એક હોય છે, અન્યથા તે શૂન્ય રાખવામાં આવે છે.) અહીં Future અને Type ચલ એ મરજિયાત છે અને તમે જોઈ શકશો કે આપણે ઉપરનાં ઉદાહરણમાં આ બંને ચલનો ઉપયોગ કરેલો નથી કારણ કે આપણે પૂરેપૂરી રકમની ચૂકવણી કરવા ઇચ્છીએ છીએ અને સમયગાળાની પહેલાં સામાન્ય રીતે આપણે હપ્તાની ચૂકવણી કરીએ છીએ, જેમાં નિષ્ફળ જતાં આપણે વધારે વ્યાજ ચૂકવવું પડે છે.

કેલ્સી નાણાં સંચાલનનાં કાર્ય માટે આ પ્રકારના અન્ય અનેક વિધેયો પૂરાં પાડે છે. ઉદાહરણોમાં વ્યાજની ગણતરી, ભવિષ્યની અમુક રકમની આજની કિંમત (net present value), ભવિષ્યની રકમ (Future value) અને રકમ પાકવાની તારીખ છે. તમે કેલ્સીની helpનો ઉપયોગ કરો અને તેના વિશે વધારે વિગત મેળવો.

નિર્ણયો લેવા (Making Decisions)

આપણે અત્યાર સુધી ચર્ચા કરેલાં વિધેયો ફક્ત કેટલાક ચલનો ઉપયોગ કરીને, તેના ઉપર પ્રક્રિયા કરીને કોઈ કિંમત પાછી આપતાં હતાં. તે ખરેખર આપણા માટે યોગ્ય કામ કરે છે પણ છતાં તે વિધેયો એટલાં ચાલાક નથી. જો વિધેય કેટલાક પ્રકારના નિર્ણયો લેવા સક્ષમ હોય તો તે વિધેય ચાલાક કહેવાય. આવા પ્રકારનાં કાર્ય માટે આપણી પાસે શરતી વિધેયો હોવાં જોઈએ કે જે પૂર્વનિર્ધારિત શરતોને આધીન નિર્ણય લઈ શકે. આ માટેનું આદર્શ ઉદાહરણ છે તમારી પરીક્ષાનું પરિણામ. નીચે આપેલું એક ઉદાહરણ લો જેમાં તમારા પરિણામની કક્ષા નક્કી કરવા માટે પૂર્વનિર્ધારિત શરતોની ચર્ચા કરેલી છે.

જો તમારા ગુણ $\geq 35\%$ અને $< 48\%$; હોય તો તમે PASS વર્ગ મેળવો છો,

જો તમારા ગુણ $\geq 48\%$ અને $< 55\%$; હોય તો તમે SECOND વર્ગ મેળવો છો,

જો તમારા ગુણ $\geq 55\%$ અને $< 60\%$; હોય તો તમે FIRST વર્ગ મેળવો છો.

આ પ્રકારના અનેક સંજોગો હોય છે જેમાં આ જાતનાં વર્ણનાત્મક તર્ક અને અગાઉથી નક્કી કરેલી શરતોની જરૂર પડે છે. કેલ્સી આ પ્રકારનાં અનેક તાર્કિક વિધેયો પૂરાં પાડે છે જે તમારા કાર્યને સરળ બનાવી દે છે. ચાલો, આપણે આ પ્રકારના કેટલાક વિધેયોનો અભ્યાસ કરીએ.

IF વિધેય વડે નિર્ણય કરવા (Making decisions with IF)

કેલ્સીમાં વપરાતા બધાં શરતી વિધેયોમાં સૌથી ધ્યાનાકર્ષક (પ્રખ્યાત) વિધેય IF છે. આ વિધેયમાં ત્રણ ચલ હોય છે : Test, True અને False. Test પદાવલિ / ચલ તમારે જે શરતની કસોટી કરવાની છે તેનો નિર્દેશ કરે છે. કેલ્સી પદાવલિની ગણતરી કરે છે અને જો પરિણામ સાચું હોય તો તે True વિભાગમાં દર્શાવેલી કિંમત પરત કરે છે અન્યથા False વિભાગમાં દર્શાવેલી કિંમત પરત કરે છે.

હવે તમે નીચે આપેલું ઉદાહરણ લો અને કેલ્સી વર્કશીટમાં પ્રયત્ન કરો :

- સેલ A2 અને B2માં અલગ અલગ કિંમત લખો.
- સેલ C2 માં જઈને નીચે આપેલું વિધેય લખો :

=IF(A2>B2,"Value at A2 is greater", "Value at B2 is greater")

- હવે સેલ A2 અને B2ની કિંમતોમાં ફેરફાર કરો અને સેલ C2માં શું ફેરફાર થાય છે તેનો અભ્યાસ કરો.

આકૃતિ 7.22માં દર્શાવ્યા પ્રમાણે રમતમાં તમારા અને તમારા મિત્રના મેળવેલા ગુણ બે સેલમાં દાખલ કરેલા હોય તો દરેક રમતમાં કોણ વિજેતા થશે તે નિર્ણય કરવા માટે એક સૂત્ર તમે લખી શકો.

B4		f(x) Σ =	=IF(B2>C2, "I win", "My friend wins")
	A	B	C
1		Me	My friend
2	Score of a game	13	17
3			
4	Result of the game:	My friend wins	

આકૃતિ 7.22 : વિજેતા કોણ છે ?

તમે ફોર્મ્યુલા બારમાં (અંડાકારથી ચિહ્નિત) સૂત્ર જોઈ શકશો. ઉપર જણાવેલાં વિદ્યાર્થીના પરિણામના ઉદાહરણમાં એક કરતાં વધારે IF વિધેયની રચના કરવાની જરૂર પડે છે. આમાં એક નીચે આપેલું છે.

ધારો કે સેલ A2 માં વિદ્યાર્થીના કુલ ગુણના ટકાનો સંગ્રહ કરેલો છે. આ માટે તમે IF વિધેય નીચે પ્રમાણે લખી શકો :

=IF(A2>=60,"First",IF(A2>=55,"Second",IF(A2>=48,"Third","Fail")))

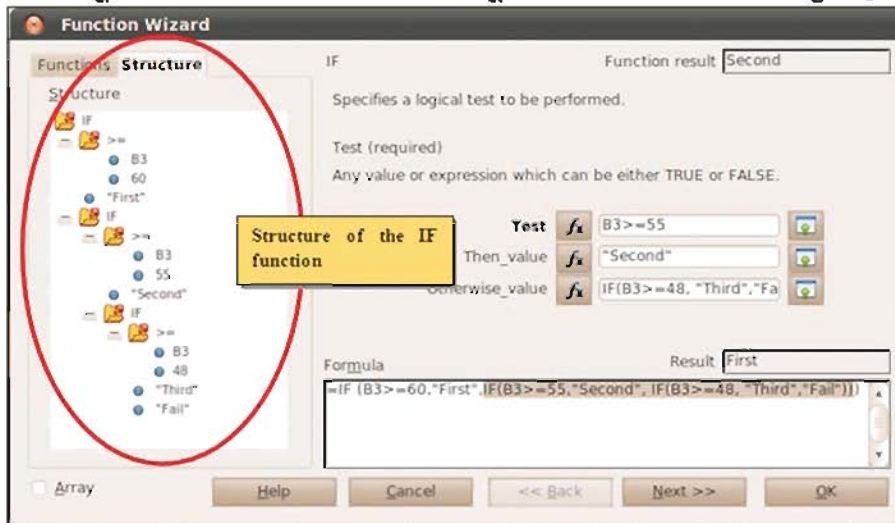
ઉપરના વિધેયમાં, પહેલું IF વિધાન સેલ A2ની કિંમત ચકાસે છે; કે જે વિદ્યાર્થીના કુલ ગુણના ટકા છે. જો સેલ A2ની કિંમત 60 બરાબર કે વધારે હશે તો પદાવલિની કિંમત સાચી ગણાશે અને વિધેય પહેલી પદાવલિમાં જણાવેલી કિંમત પાછી આપશે, કે જે "First" છે. જો વિદ્યાર્થીનાં 60 કરતાં ઓછા ટકા હશે તો તે વિધેયની બીજી પદાવલિની કિંમત આપશે કે જે અન્ય IF વિધાન છે. આ રીતે તમે એક IFની અંદર અનેક IFનો સમાવેશ કરી શકો અને આથી તેને nested if કહેવામાં આવે છે. આકૃતિ 7.23માં આ દર્શાવેલું છે. આકૃતિમાં IF વિધાનની જુદી જુદી શક્યતાઓ દર્શાવવા માટે આપણે અનેક વિદ્યાર્થીઓનાં નામ અને તેણે મેળવેલા ટકા આપેલા છે. જો કે આપણે સૂત્ર ફક્ત એક જ વાર લખેલું છે. (જુઓ ફોર્મ્યુલા બાર ઉપર નિશાની કરેલ છે) અને આકૃતિ 7.23માં દર્શાવ્યા પ્રમાણે તે સૂત્રને અન્ય વિદ્યાર્થીઓ માટે ડ્રોગ (નકલ) કરેલું છે.

	A	B	C	D	E	F	G
1							
2	Name of student:	Mita	Sita	Rita	Gita	Nita	Babita
3	Student's marks in %:	77	69	59	49	44	30
4							
5	Class of the student:	First	First	Second	Third	Fail	Fail
6							
7							

Copying the formula

આકૃતિ 7.23 : IF વિધેયનું ઉદાહરણ

જ્યારે મોટા સૂત્ર લખવાનાં હોય છે ત્યારે તેના ઉપરની પકડ ઢીલી થઈ જવાની શક્યતા વધી જાય છે અને તે લખવામાં ભૂલ થઈ શકે છે. આ ટાળવા માટે તમે ફંક્શન વિઝાર્ડની મદદ લઈ શકો છો. ફંક્શન વિઝાર્ડ વાપરવાનો એક ફાયદો એ છે કે તમારે વિધેયનું સર્વથા ખરું નામ ટાઈપ કરવું પડતું નથી. એક વખત તમે ફંક્શન વિઝાર્ડ જુઓ અને તેને વાપરી જુઓ, તમે Structure ટેબ પસંદ કરો તેથી આકૃતિ 7.24માં દર્શાવ્યા પ્રમાણે તમે લખેલા સૂત્રનું બંધારણ જોવા મળશે કે જે વિધેય બનાવવાનું બંધારણ આબેહૂબ રીતે પ્રદર્શિત કરે છે. જો તેમાં કંઈ ભૂલો હશે તો તે યાદીમાં લાલ બિંદુ સ્વરૂપે દર્શાવે છે.



આકૃતિ 7.24 : વિધેયનું બંધારણ

સંબંધિત પ્રક્રિયકો (Relational operators)

ઉપરનાં ઉદાહરણમાં જ્યારે આપણે IF વિધાનની ચર્ચા કરતા હતા ત્યારે આપણે >= જેવાં પ્રક્રિયક (ઓપરેટર) જોયાં હતાં. આવાં પ્રક્રિયકો (ઓપરેટર્સ) રિલેશનલ ઓપરેટર કહેવાય છે. કોષ્ટક 7.1માં તમારે જરૂરી રિલેશનલ ઓપરેટરની યાદી આપેલી છે.

રિલેશનલ ઓપરેટર	ચિહ્ન	વર્ણન
Equal	=	બંને સંકાર્ય (operands) (જેમ કે A1 અને A2) સરખા છે. ઉદાહરણ =IF(A1=A2, "A1 is equal to A2", "A1 and A2 not equal")
Less than	<	પ્રથમ સંકાર્ય બીજા સંકાર્ય કરતાં નાનો છે. ઉદાહરણ =IF(A1<A2, "A1 is less than A2", "A1 is not less than A2")
Greater than	>	પ્રથમ સંકાર્ય બીજા સંકાર્ય કરતાં મોટો છે. ઉદાહરણ : =IF(A1>A2, "A1 is greater than A2", "A1 is not greater than A2")
Less than or equal to	<=	પ્રથમ સંકાર્ય બીજા સંકાર્ય કરતાં નાનો અથવા બરાબર છે. ઉદાહરણ : =IF(A1<=A2, "A1 is less than or equal to A2", "A1 is not less than or equal to A2")
Greater than or equal to	>=	પ્રથમ સંકાર્ય બીજા સંકાર્ય કરતાં મોટો અથવા બરાબર છે. ઉદાહરણ: =IF(A1>=A2, "A1 is greater than or equal to A2", "A1 is not greater than or equal to A2")
Different from	<>	બંને સંકાર્ય સરખા નથી, પ્રથમ સંકાર્ય બીજા સંકાર્ય બરાબર નથી. ઉદાહરણ : =IF(A1<>A2, "A1 is different than A2", "A1 is not different than A2")

કોષ્ટક 7.1 : વિવિધ રિલેશનલ ઓપરેટર (સંબંધિત પ્રક્રિયકો)

તાર્કિક વિધેયો (Logical functions)

કેલ્સી અન્ય કેટલાંક વિધેયો પૂરાં પાડે છે જે તાર્કિક નિર્ણયો લેવામાં મદદરૂપ થાય. આ વિધેયોમાં AND, OR અને NOTનો સમાવેશ થાય છે. આ વિધેયો અટપટાં સૂત્રો અને અન્ય વિધેયોની રચના કરવામાં ઉપયોગી બને છે. આ વિભાગમાં ઉપર જણાવેલા ત્રણ તાર્કિક વિધેયોની સમજ આપેલી છે.

TRUE

આ વિધેય આપણને જે કિંમત પાછી આપે છે, તે તાર્કિક કિંમત TRUE (ખરું) છે. આ વિધેયની વાક્યરચના **TRUEQ** છે.

FALSE

આ વિધેય આપણને જે કિંમત પાછી આપે છે, તે તાર્કિક કિંમત FALSE છે. આ વિધેયની વાક્યરચના **FALSEQ** છે.

NOT

આ વિધેયની વાક્યરચના **NOT (પદાવલિ કે જેનું પરિણામ તાર્કિક કિંમતમાં આવે)** છે. આ NOT વિધેય તાર્કિક કિંમતને તેનાથી વિપરીત બનાવે છે. આથી FALSE બદલાય ને TRUE બને છે અને TRUE બદલાય ને FALSE બને છે. નીચેનાં ઉદાહરણો જુઓ :

=NOT(6<>6)ની કિંમત TRUE મળે છે.

=NOT(1=1)ની કિંમત FALSE મળે છે.

આપણે NOT(TRUE) અને NOT(FALSE) લખી શકીએ ? પ્રયત્ન કરી જુઓ. આકૃતિ 7.25માં TRUE, FALSE અને NOT (FALSE) નાં કેટલાંક ઉદાહરણો આપેલાં છે.

	A	B
1		
2	Logical function	Value
3	TRUE()	TRUE
4	FALSE()	FALSE
5	NOT (TRUE)	FALSE
6	NOT (FALSE)	TRUE
7	NOT (6<>6)	TRUE
8	NOT (1=1)	FALSE
9		

આકૃતિ 7.25 : TRUE, FALSE અને NOT વિધેયોનાં ઉદાહરણો

AND વિધેય

આ વિધેય વધારેમાં વધારે 30 ચલ(arguments)નાં પરિણામોને સરખાવવા માટે વપરાય છે. જો બધી જ શરતો / ચલનું પરિણામ TRUE આવશે તો જ ANDનું પરિણામ TRUE મળશે. આ વિધેયની વાક્યરચના **AND(Cond_1,Cond_2,...,Cond_30)** છે.

અહીં AND વિધેયના બધા ચલ એ શરતો છે જેનું પરિણામ TRUE અથવા FALSE માં આવી શકે, એટલે કે બૂલિયન કિંમતમાં (Boolean values) જવાબ મળે. હવે આ ઉદાહરણ લો. જો આપણે જાણવું હોય કે સેલ A1, A2 અને A3ની કિંમતો સરખી છે કે નહીં તો આપણે એ જાણવું જોઈએ કે A1 અને A2 સરખા છે કે નહીં અને પછી આપણે જાણીએ કે A2 અને A3 સરખા છે કે નહીં. જો બંનેના પરિણામ સાચાં હોય તો આપણે નક્કી કરી શકીએ કે ત્રણેની કિંમત સરખી છે. એટલે કે, આપણે "=" રિલેશનલ ઓપરેટર વાપરીને બે શરતો બનાવવી પડે કે જે A1=A2 અને A2=A3 છે. આથી, AND વિધેય **=AND(A1=A2,A2=A3)** લખી શકાય. ઉપર જે ચર્ચા કરી તે આકૃતિ 7.26માં દર્શાવેલી છે.

C5	f(x)	Σ	=	=AND(A1=A2,A2=A3)
	A	B	C	
1	12			
2	12			
3	6			
4				
5	Are the above values same? :			FALSE
6				

આકૃતિ 7.26 : AND વિધેયનું ઉદાહરણ

OR વિધેય

આ વિધેયની વાક્યરચના **OR(Cond_1,Cond_2,...,Cond_30)** છે.

આ OR વિધેયમાં વધારેમાં વધારે 30 શરતોના (arguments)નાં પરિણામોને સરખાવવા માટે થાય છે. જો આપેલી શરતોમાંથી ઓછામાં ઓછી એક શરતનું પરિણામ TRUE હશે તો OR આપણને TRUE કિંમત પાછી આપશે. ઉદાહરણ તરીકે, આપણે ઉપર વર્ણવેલા તાર્કિક AND વિધેયનું સૂત્ર લઈએ. આપણે સૂત્રમાં થોડો ફેરફાર કરીએ. અગાઉ આપણે ત્રણ ચલ A1, A2 અને A3 સરખા છે તેમ ઈચ્છતા હતા. હવે આપણે ત્રણમાંથી કોઈ પણ બે ચલની કિંમત સરખી હોય તો આપણે આગળ વધવા ઈચ્છીએ છીએ. આ માટે **=OR(A1=A2,A2=A3,A1=A3)** લખી શકાય.

નીચેનાં સંજોગોમાં ઉપરનું વિધેય TRUE કિંમત પાછી આપશે :

- જો A1 બરાબર A2 હોય;
- જો A2 બરાબર A3 હોય;
- જો A1 બરાબર A3 હોય; અથવા
- જો A1, A2 અને A3 ની કિંમત સરખી હોય.

જો ઉપરના બધાં જ વિધાનો ખોટાં હોય તો વિધાન FALSE કિંમત પાછી આપશે. એટલે કે નીચેનાં બધાં વિધાનો સાચા હોવાના સંજોગોમાં વિધાન FALSE કિંમત આપશે.

- A1 બરાબર A2 નથી; A2 બરાબર A3 નથી; અને A1 બરાબર A3 નથી; જો આ બધા વિધાનો સાચાં હોય તો જ પરિણામ FALSE મળશે.

એટલે કે જો વિધેયમાં આપેલી બધી શરતો FALSE હોય તો જ OR વિધેય FALSE કિંમત પાછી આપશે. આકૃતિ 7.27માં OR વિધેયનાં કેટલાંક ઉદાહરણો આપેલાં છે અને કોષ્ટક 7.2માં તાર્કિક વિધેયનાં કેટલાંક ઉદાહરણો આપેલાં છે.

A4	f(x)	Σ	=	=OR(A1=A2,A2=A3,A1=A3)		
	A	B	C	D	E	
1	12	12	12	6	6	
2	12	12	6	12	12	
3	12	6	12	12	31	
4	TRUE	TRUE	TRUE	TRUE	FALSE	

આકૃતિ 7.27 : OR વિધેયનાં ઉદાહરણ

ઉદાહરણો			પદાવલિની કિંમત	પરિણામ
A1ની કિંમત	A2ની કિંમત	A3ની કિંમત		
12	12	12	A1=A2 → TRUE A2=A3 → TRUE A1=A3 → TRUE	TRUE
12	12	6	A1=A2 → TRUE A2=A3 → FALSE A1=A3 → FALSE	TRUE
12	6	12	A1=A2 → FALSE A2=A3 → FALSE A1=A3 → TRUE	TRUE
6	12	12	A1=A2 → FALSE A2=A3 → TRUE A1=A3 → FALSE	TRUE
6	12	31	A1=A2 → FALSE A2=A3 → FALSE A1=A3 → FALSE	FALSE

કોષ્ટક 7.2 : તાર્કિક વિધેય OR(A1=A2;A2=A3;A1=A3)નાં વિવિધ ઉદાહરણો

સ્ટ્રિંગ ફંક્શન (String functions) – શબ્દિક લખાણ ઉપરનાં વિધેયો

શબ્દિક લખાણની બે સ્ટ્રિંગ જોડવી (Joining two strings of text)

CONCATENATE વિધેય શબ્દિક લખાણની વધારેમાં વધારે 30 સ્ટ્રિંગ (string) ને જોડી શકે છે. આ વિધેયની વાક્યરચના **CONCATENATE(text1,text2,...,text30)** છે.

તમે CONCATENATE વિધેયનાં નામને બદલે '&' ચિહ્ન પણ વાપરી શકો છો. એટલે કે "ice" અને "cream" જેવી બે સ્ટ્રિંગને જોડવા માટે તમે વિધેયને નીચે મુજબ લખી શકો :

= "ice" & "cream" અથવા

=CONCATENATE("ice","cream")

ઉપરનાં બંને વિધેયોનું પરિણામ સ્ટ્રિંગ "icecream" મળશે. જો બંને શબ્દો વચ્ચે એક ખાલી જગ્યા (space) તમારે જોઈતી હોય તો તમે બેમાંથી કોઈ પણ એક સ્ટ્રિંગમાં તમે ઉમેરી શકો જેમ કે **= "ice " & "cream" અથવા**
= "ice" & " cream"

જે સ્ટ્રિંગ તમારે જોડવાની છે તેનો સંગ્રહ અલગ અલગ સેલ(ધારો કે A1 અને A2)માં કરેલો હોય તો તમે **=A1&A2** અથવા **=CONCATENATE(A1,A2)** પણ લખી શકો.

નીચે આપેલો સ્વાધ્યાય કરો :

- સેલ A1 ઉપર જઈ તમારું પ્રથમ નામ લખો.
- સેલ A2 ઉપર જઈ તમારા પિતાનું નામ લખો.
- સેલ A3 ઉપર જઈ તમારી અડક લખો.
- સેલ A4 ઉપર જઈ એક ખાલી જગ્યા " " લખો.
- & વિધેયનો ઉપયોગ કરી સેલ C3માં **=A1&A4&A2&A4&A3**

આકૃતિ 7.28માં CONCATENATION વિધેયનાં કેટલાંક ઉદાહરણો આપેલાં છે.

A4	$f(x)$	Σ	=	=CONCATENATE(B1,B2)
	A	B		C
1		ice		
2	+	cream		
3	-----			
4	icecream			
5				
6	Use of & operator			
7	"ice"&"cream"			icecream
8	"ice" & " " & "cream"			ice cream
9	B1 & " " & B2			ice cream
10	B1 & B2			icecream

આકૃતિ 7.28 : CONCATENATE વિધેયનાં ઉદાહરણો

LOWER વિધેય

આ વિધેય આપેલા બધા અક્ષરોને નાના મૂળાક્ષરો(lower case)માં ફેરવે છે. આ વિધેયની વાક્યરચના **LOWER (શબ્દિક લખાણ)** છે. ઉદાહરણ : **=LOWER("School")** આપણને જવાબમાં કિંમત "school" પરત આપે છે.

Upper વિધેય

આ વિધેય આપેલા બધા અક્ષરોને મોટા મૂળાક્ષરો(કિપિટલ અક્ષરો upper case)માં ફેરવે છે. આ વિધેયની વાક્યરચના **UPPER(text)** છે. ઉદાહરણ : **=UPPER("School")** આપણને જવાબમાં કિંમત "SCHOOL" આપે છે.

Proper વિધેય

આ વિધેય બધા શબ્દોનો પહેલો અક્ષર મોટા મૂળાક્ષરમાં અને બાકીના અક્ષરો નાના મૂળાક્ષરોમાં ફેરવે છે. આ વિધેયની વાક્યરચના **PROPER(text)** છે. ઉદાહરણ : **=PROPER ("I love my sChool")** ની કિંમત **I Love My School** પાછી આપે છે.

સામાન્ય રીતે LOWER, UPPER અને PROPER વિધેયો દસ્તાવેજને આકર્ષક સ્વરૂપ આપવા માટે (ફોર્મેટ કરવા માટે) વપરાય છે. યોગ્ય રીતે ફોર્મેટ કરેલા દસ્તાવેજની સુવાચ્યતા અને સ્પષ્ટતા ઘણી સારી હોય છે.

રોમન અને અરેબિક અંક (Roman and Arabic) માટેનાં વિધેય

ROMAN વિધેય આપેલી સંખ્યાને રોમન અંકમાં ફેરવે છે. તેની વાક્યરચના **ROMAN(number, mode)** છે.

અહીં એ નોંધ કરો કે 0થી 3999ની સંખ્યા જ રોમન અંકમાં રજૂ કરી શકાય. આથી આપણે ચલમાં આપેલી સંખ્યાની કિંમત આ હદમાં જ હોવી જોઈએ. તમે જાણો છો કે રોમન સંખ્યા ઘણી જુદી જુદી રીતે લખી શકાય છે. ધારો કે સંખ્યા 8ને તમારે રોમનમાં લખવી છે. તમે આ સંખ્યા VIIIની રીતે લખી શકો, જો કે આ જ સંખ્યાને તમે IIX તરીકે પણ લખી શકો. વિધેયમાંનું બીજું ચલ આ રીતે રજૂઆતના પ્રકાર માટે વપરાય છે. આ Mode ચલની કિંમત 0, 1, 2 અથવા 3 હોઈ શકે. TV અને ફિલ્મ ઉદ્યોગમાં ફિલ્મની પટ્ટી અને બીજી વસ્તુઓને સાંકેતિક ભાષામાં રાખવા માટે રોમન સંખ્યાનો ઉપયોગ થાય છે.

અરેબિક વિધેય રોમન વિધેય કરતાં ઊલટું છે, તે રોમન સંખ્યાને તેના ચલ તરીકે લઈને તેને અરેબિક સંખ્યામાં ફેરવે છે. નીચે આપેલું ઉદાહરણ જુઓ :

=ROMAN (125) ની કિંમત **CXXV** પાછી મળે છે.

વિધેય **=ARABIC("MMXI")** શું કિંમત પાછી આપે છે તે શોધી કાઢો ? તે **2013** છે ?

આકૃતિ 7.29માં કેટલાંક ROMAN અને ARABIC વિધેયનાં ઉદાહરણો આપેલાં છે.

	A	B	C
1	Function	Value	
2	ROMAN(125)	CXXV	
3	ARABIC("mmxiii")	2013	
4	ARABIC(ROMAN(2013))	2013	
5			
6	Also consider following examples to see how mode works		
7	ROMAN(999)	CMXCIX	
8	ROMAN(999,0)	CMXCIX	
9	ROMAN(999,1)	LMVLIV	
10	ROMAN(999,2)	XMIX	
11	ROMAN(999,3)	VMIV	

આકૃતિ 7.29 : ROMAN અને ARABIC વિધેયનાં ઉદાહરણો

Trim વિધેય

આ વિધેય આપેલાં શાબ્દિક લખાણમાંથી (સ્ટ્રિંગમાંથી) વધારાની ખાલી જગ્યા (spaces) દૂર કરે છે. એટલે કે જો તમે **TRIM("I Love My School!")** લખો તો તે તમને **I Love My School!** પાછું આપશે. આ વિધેયની વાક્યરચના **TRIM(text)** છે.

સ્ટ્રિંગની સરખામણી કરવી (Comparing Strings)

EXACT વિધેય આપેલી બે સ્ટ્રિંગની સરખામણી કરે છે અને જો તે આખી સ્ટ્રિંગ બિલકુલ સરખી હોય તો 1 કિંમત પરત કરે છે, અને જો સરખી ન હોય તો 0 કિંમત પરત કરે છે. ઉદાહરણ : **=EXACT("Blue", "Blue")** આપણને 1 કિંમત પરત કરે છે જ્યારે **=EXACT("Blu", "Blue")** આપણને 0 કિંમત પરત કરે છે.

આ વિધેયની વાક્યરચના **EXACT(s1, s2)** છે.

સ્ટ્રિંગના પેટાભાગ પાડવા (Substring)

કેલ્સીમાં ડાબી બાજુથી સ્ટ્રિંગનો પેટાભાગ કરવા માટે અને બીજુ જમણી બાજુથી સ્ટ્રિંગનો ભાગ કરવા માટે એમ બે વિધેયો ઉપલબ્ધ છે.

LEFT વિધેય શાબ્દિક લખાણમાંથી ડાબી બાજુનો પહેલો અક્ષર અથવા આપણે જણાવેલા n અક્ષરો પાછા આપે છે.

આ વિધેયની વાક્યરચના **LEFT(Text, n)** છે. આથી, આપેલી સ્ટ્રિંગની ડાબી બાજુથી 4 અક્ષરો જુદા પાડવા માટે આપણે **=LEFT("Ram Dhashrathbhai Patel", 4)** લખી શકીએ, જે **"Ram "** સ્ટ્રિંગ પાછી આપશે.

આ જ પ્રમાણે RIGHT વિધેય આપણને આપેલા શાબ્દિક લખાણની સ્ટ્રિંગમાંથી જમણી બાજુના છેડાથી n અક્ષરો છૂટા પાડી જવાબ પાછો આપે છે. એટલે કે જો આપણે **=RIGHT("Ram Dhashrathbhai Patel", 5)** લખીએ તો **"Patel"** જવાબ મળશે. આ વિધેયની વાક્યરચના **RIGHT(Text, n)** છે.

કેલ્સી સ્ટ્રિંગની વચ્ચેથી લખાણ જુદું કરવાની સગવડતા પણ પૂરી પાડે છે. આ વિધેયની વાક્યરચના **MID(Text, Start, n)** છે.

ઉપરનાં ઉદાહરણમાં આપેલાં નામમાંથી વચ્ચેનું નામ જુદું પાડવા માટે આપણે **=MID("Ram Dhashrathbhai Patel", 5, 13)** લખીએ તો **"Dhashrathbhai"** સ્ટ્રિંગ પાછી મળશે.

સ્ટ્રિંગની લંબાઈ (Length of a String) LEN વિધેય

LEN વિધેય આપેલી (શાબ્દિક લખાણ) સ્ટ્રિંગની લંબાઈ (કેટલા અક્ષરનો સમાવેશ છે તે) જણાવે છે. તેની વાક્યરચના **LEN(Text)** છે. આથી જો આપણે **LEN("Ram")** લખીએ તો આપણને 3 જવાબ મળશે. આકૃતિ 7.30માં સ્ટ્રિંગને લગતાં વિધેયોનાં ઉદાહરણો આપેલાં છે.

	A	B
1	Function	Value
2	TRIM("I Love My School")	I Love My School
3	EXACT("Blue", "Blue")	TRUE
4	EXACT("My School", "New School ")	FALSE
5	LEFT("Ram D Patel", 3)	Ram
6	RIGHT("Ram D Patel", 5)	Patel
7	MID("Ram D Patel", 5,1)	D
8	LEN("Ram D Patel")	11

આકૃતિ 7.30 : સ્ટ્રિંગને લગતાં વિધેયોનાં ઉદાહરણો

Time અને Date વિધેયો (Time and Date functions)

કેલ્સી આપણને સમય અને તારીખની ગણતરી માટેના વિધેય પૂરા પાડે છે. જેમાં નાનો સમયગાળો (જેમ કે કલાક, મિનિટ અથવા સેકન્ડમાં) હોય તો તમે time વિધેયનો ઉપયોગ કરી શકો અને મોટા સમયગાળા માટે (જેમ કે અઠવાડિયા, મહિનાઓ અથવા વર્ષો) તમે Date વિધેયનો ઉપયોગ કરી શકો.

આજની તારીખ જાણવા માટે (Today's date) TODAY વિધેય

TODAY વિધેય તમારા કમ્પ્યુટરમાં સંગ્રહ કરેલી આજની તારીખ જણાવે છે. આ વિધેયમાં કોઈ ચલ નથી. ધારો કે આજની તારીખ 11મી જુલાઈ 2013 છે તો TODAY() વિધેય આપણને 11/07/2013 અથવા 11. Jul. 2013 (તમારા કમ્પ્યુટરમાં જે રીતે ફોર્મેટ સેટ કરેલું હશે તે પ્રમાણે) જવાબ આપશે.

Date વિધેય

DATE વિધેય આપણે આપેલાં વર્ષ, તારીખ અને દિવસ પ્રમાણે માન્ય તારીખ આપે છે. એટલે કે જો તમે **=DATE(2013, 1, 1)** આપો તો તે વિધેય 1/1/13 કિંમત પાછી આપશે.

Day વિધેય

DAY વિધેય કોઈ પણ માન્ય સ્વરૂપમાં આપેલી તારીખને મહિનાના દિવસમાં ફેરવવા માટે વપરાય છે. ઉદાહરણ : જો આજે 11મી જુલાઈ 2013 હોય તો **=DAY(today())** આપણને 11 જવાબ આપશે.

ઉપર જણાવેલું વિધેય આપણે **=DAY(DATE(2013,7,11))** પણ લખી શકીએ. આ વિધેયની વાક્યરચના **DAY(Date)** છે.

DATE વિધેયમાં આપેલું ચલ તારીખ સ્વરૂપમાં પૂરું પાડવાની ખાતરી માટે આપણે અગાઉ વાપર્યું હતું તે રીતે DATE વિધેય વાપરી શકો. DATE વિધેય સૌપ્રથમ આપેલી સંખ્યાને માન્ય તારીખનાં સ્વરૂપમાં ફેરવે છે અને પછી તારીખમાંથી દિવસનો નંબર પાછો આપે છે.

WEEKDAY વિધેય

WEEKDAY વિધેય આપણે આપેલી સાચી તારીખને અઠવાડિયાના વાર (જેમ કે સોમવાર, મંગળવાર,...) માં ફેરવે છે. દિવસોને અનુક્રમ નંબર 0 (રવિવાર)થી 6 (શનિવાર) આપવામાં આવે છે. ઉદાહરણ : જો તમે **=WEEKDAY(DATE(2013,7,11))** લખો તો જવાબ 5 મળશે.

આ વિધેયની વાક્યરચના WEEKDAY(Date, n) છે. અહીં Date એ કોઈ માન્ય તારીખ છે અને n એ સંખ્યા છે જે અઠવાડિયું કયા દિવસથી શરૂ થાય છે તે જણાવે છે. તમે n માટે નીચેની કિંમતો વાપરી શકો છો.

N=1 : અઠવાડિયું રવિવારથી શરૂ થાય છે. (દિવસ 0).

N=2 : અઠવાડિયું સોમવારથી શરૂ થાય છે. (દિવસ 1).

જો તમે ચલ (argument) ન જણાવો તો તે કિંમત 1 લેશે, એટલે કે ઉપરના ઉદાહરણમાં જણાવ્યા પ્રમાણે દિવસોને અનુક્રમ નંબર 0 (રવિવાર)થી 6 (શનિવાર) ગણાશે.

નીચે કેટલાંક ઉદાહરણો આપ્યાં છે :

=WEEKDAY(DATE(2013,7,11),0)નો જવાબ 3 મળશે.

=WEEKDAY(DATE(2013,7,11),1)નો જવાબ 5 (પૂર્વનિર્ધારિત) કે જે

=WEEKDAY(DATE(2013,7,11)) બરાબર થાય.

=WEEKDAY(DATE(2013,7,11),2)નો જવાબ 4 મળશે.

Month વિધેય

MONTH વિધેય કોઈ માન્ય તારીખને મહિનાના નંબરમાં ફેરવે છે. ઉદાહરણ : જો તમે **=MONTH(DATE(2013,7,11))** કોઈ સેલમાં લખશો તો જવાબ 7 મળશે.

આ વિધેયની વાક્યરચના **MONTH(t)** છે.

YEAR વિધેય

YEAR વિધેય માન્ય તારીખને વર્ષમાં ફેરવે છે. ઉદાહરણ તરીકે, જો તમે **=YEAR(DATE(2013,7,11))** કોઈ સેલમાં લખશો તો જવાબ 2013 મળશે.

આ વિધેયની વાક્યરચના **YEAR(t)** છે.

વર્ષમાં દિવસોની સંખ્યા (Number of Days in a Year)

DAYSINYEAR વિધેય ચલની તારીખ આપેલ હોય તે વર્ષના કુલ દિવસોની સંખ્યા આપે છે. સ્પ્રેડશીટના સેલમાં નીચેનું વિધેય લખો :

=DAYSINYEAR(DATE(2013,7,11)).

આ વિધેય 365 જવાબ આપશે.

આ વિધેયની વાક્યરચના **DAYSINYEAR(t)** છે.

આપેલું વર્ષ લીપ યર છે કે કેમ તે જાણવા માટે કદાચ તમે અગાઉ કોઈ વિશિષ્ટ પ્રોગ્રામ કે ગૂંચવાડાભર્યું સૂત્ર લખ્યું હશે. એના બદલે તમે વર્ષમાં કેટલા દિવસ છે તે ચકાસીને જાણી શકો છો.

બે તારીખનો તફાવત (Difference between two dates) – DAYS વિધેય

DAYS વિધેય આપેલી બે તારીખ વચ્ચેના દિવસોની ગણતરી કરે છે. અહીં એ નોંધ કરો કે વિધેયનું નામ DAYને બદલે DAYS છે. અગાઉ ચર્ચા કરેલું DAY વિધેય મહિનાનો કેટલામો દિવસ છે તે જણાવે છે. DAYS વિધેય આપણે આપેલી બે માન્ય તારીખો વચ્ચેના દિવસોની સંખ્યા આપે છે.

જો આપણે **=DAYS(DATE(2012,7,11), DATE(2013,7,11))** લખીએ તો 366 જવાબ મળશે. 2012 વર્ષ લીપ યર હોવાથી તેમાં 366 દિવસો છે !

હવે આ વિધેય નીચે દર્શાવ્યા પ્રમાણે વાપરો :

=DAYS(A1, NOW())

અહીં A1 એક સેલ સ્થાનાંક છે જેમાં તમારે તમારી જન્મ તારીખ લખવાની છે. બીજો ચલ NOW() છે, જે કમ્પ્યુટરમાં સંગ્રહ કરેલી આજની તારીખ સમય સાથે પાછી આપશે. હવે વિચારો કે ઉપર જણાવેલું વિધેય શું જવાબ આપશે ? બરાબર છે, તમે સાચા છો, તે તમારી ઉંમર દિવસોમાં જણાવશે. આ રીતે નિશ્ચિતપણે તમે શોધી શકશો કે તમારા વર્ગમાં સૌથી મોટું કોણ છે ?

NOW વિધેયની વાક્યરચના **NOW()** જ છે. આ વિધેયમાં કોઈ ચલ નથી.

અહીં નોંધ કરો કે બીજી તારીખ પહેલી તારીખમાંથી બાદ કરવામાં આવે છે, આથી ઘણા સમયે વિધેય ઋણ કિંમત પાછી આપશે. આ ટાળવા માટે તમે ABS વિધેયનો ઉપયોગ કરી શકો. આકૃતિ 7.31માં તારીખ અને સમયને લગતા કેટલાક વિધેયોનાં ઉદાહરણો આપેલાં છે.

	A	B
1	Function	Value
2	TODAY()	11. Jul. 2013
3	DATE(2013,7,11)	July 11, 2013
4	DAY(DATE(2013,7,11))	11
5	WEEKDAY(DATE(2013,7,11))	5
6	WEEKDAY(DATE(2013,7,11),0)	3
7	WEEKDAY(DATE(2013,7,11),1)	5
8	WEEKDAY(DATE(2013,7,11),2)	4
9	MONTH(DATE(2013,7,11))	7
10	YEAR(DATE(2013,7,11))	2013
11	DAYSINYEAR(DATE(2013,7,11))	365
12	DAYS(DATE(2012,7,11), DATE(2011,7,11))	366
13	WEEKS(DATE(2011,7,11), DATE(2012,7,11),0)	52
14	YEARS(DATE(2011,7,11), DATE(2013,7,11),0)	2

આકૃતિ 7.31 : તારીખ અને સમયને લગતાં વિધેયો

આપેલી બે તારીખો વચ્ચે અઠવાડિયાની સંખ્યા (Number of weeks between two dates)

આપણે જાણીએ છીએ કે DAY અને DAYS નામનાં બે વિધેયો છે; એ જ રીતે WEEKS બે તારીખો વચ્ચે અઠવાડિયાઓની સંખ્યાની ગણતરી કરે છે.

=WEEKS(DATE(2011,7,11), DATE(2012,7,11),0)નો જવાબ 52 મળશે.

ઉપર જણાવેલાં વિધેયમાં બે DATE વિધેયોનો ઉપયોગ કરેલો છે જે આપેલા ચલને માન્ય તારીખના સ્વરૂપમાં ફેરવે છે. આ ઉપરાંત, WEEKS વિધેય એક વધારે ચલ '0' નો પણ ઉપયોગ કરે છે, કે જે અઠવાડિયાના પ્રકારનો નિર્દેશ કરે છે. આ ચલની '0' ક્રિમત અઠવાડિયાનો સમયગાળો જણાવે છે. અહીં તમે '0' ને બદલે '1' જણાવીને કેલેન્ડર અઠવાડિયાનો ઉપયોગ કરી શકો છો.

WEEK વિધેયની વાક્યરચના **WEEKS(t1, t2, Week type)** છે. જેવી રીતે DAY અને DAYS બે અલગ અલગ વિધેયો છે એ જ રીતે WEEK અને WEEKS બે અલગ અલગ વિધેયો છે.

આપેલી બે તારીખો વચ્ચે વર્ષની સંખ્યા (Number of years between two dates)

YEARS વિધેય આપેલી બે માન્ય તારીખો વચ્ચે વર્ષની સંખ્યાની ગણતરી કરે છે કે જે વર્ષના સમયગાળામાં (ચલની ક્રિમત 0 જણાવીને) અથવા કેલેન્ડર વર્ષમાં (ચલની ક્રિમત 1 જણાવીને) હોય છે. ઉદાહરણ તરીકે, જો તમે

=YEARS(DATE(2011,7,11), DATE(2013,7,11), 0) લખશો તો તમે 2 જવાબ મેળવશો.

આ વિધેય વડે તમે કેટલાં વર્ષનાં થયાં તે જાણી શકશો. તમારી જન્મ તારીખથી આજની તારીખ વચ્ચે કેટલાં વર્ષ છે તે ફક્ત શોધી કાઢો.

અન્ય સ્પ્રેડશીટ પેકેજમાં ઉપલબ્ધ વિધેયો (Functions in other Spreadsheet Packages)

ઘણા ખરા સ્પ્રેડશીટ પેકેજ આ જ પ્રકારનાં વિધેય પૂરા પાડે છે. જો તમે માઈક્રોસોફ્ટ એક્સલનું ઉદાહરણ લેશો તો તમને જણાશે કે તે પણ આ પ્રકરણમાં ચર્ચા કરી તેમાંના મોટાભાગના વિધેયો પૂરા પાડે છે.

ક્યારેક તમારે તેની વાક્યરચના (syntax)માં ગૌણ પરિવર્તન કરવા પડે છે જેમ કે તેના સ્પેલિંગમાં કોઈ નાનો ફેરફાર અથવા વિધેયના ચલ વચ્ચે ';' (અર્ધવિરામ) કે ',' (અલ્પવિરામ)નો ઉપયોગ.

ગૂગલ સ્પ્રેડશીટ પેકેજ [doc.google.com] પણ આ જ પ્રકારનાં વિધેયો પૂરા પાડે છે. તમે જાણો છો કે તે ઇન્ટરનેટ, મોબાઈલ ફોન અને એન્ડ્રોઈડ જેવી ચાલક પદ્ધતિ વાપરતાં યંત્રોમાં ખૂબ ઉપયોગી થાય છે.

આ ઉપરાંત ગૂગલ પેકેજ તમને નિઃશુલ્ક સ્ટ્રોડશીટ ટેમ્પલેટ પણ અનેક કાર્યો કરવા - જેમ કે વિદ્યાર્થીઓનું સમયપત્રક, પ્રગતિપત્રકનો અહેવાલ, પ્રોજેક્ટનું આયોજન, વિદ્યાર્થીઓ અને કર્મચારીઓની હાજરી, અંગત આવક-જાવકનું નિયમન, લગ્નપ્રસંગ માટે મહેમાનોની યાદી, માલ-વેચાણનું ભરતિયું બનાવવું વગેરે માટે પૂરાં પાડે છે.

આ ટેમ્પલેટમાં તૈયાર માળખાં જેમ કે વિભાગો, ઊભી હરોળ અને આડી હરોળનાં શીર્ષકો, જરૂરી ગણતરીઓ કરવા માટેનાં સૂત્રો તેમજ સ્ક્રિપ્ટ હોય છે. ગૂગલ સ્ટ્રોડશીટ જેવાં વિનિયોગો પ્લેટફોર્મથી સ્વતંત્ર અને અન્ય જગ્યાએ ફેરવવા સરળ હોવાથી ખૂબ જ પ્રચલિત બની રહ્યા છે.

સારાંશ

આપણે આ પ્રકરણમાં વિવિધ ગાણિતિક, આંકડાકીય, સ્ટ્રિંગ, તારીખ અને સમયને લગતાં તેમજ ફોર્મેટિંગ માટેનાં વિધેયોનો અભ્યાસ કર્યો. આપણે ફંક્શન વિઝાર્ડ મારફતે કઈ રીતે વિધેય દાખલ કરી શકાય તે પણ જાણ્યું. આ ઉપરાંત, વિધેયો સાથે કાર્ય કરતા સમયે ભૂલો તરફ નિર્દેશ કરવામાં અને તેને સુધારવામાં ફંક્શન વિઝાર્ડ મદદરૂપ થાય છે તે પણ શીખ્યા.

સ્વાધ્યાય

1. ફંક્શન વિઝાર્ડનું કાર્ય ટૂંકમાં સમજાવો.
2. તમારી પસંદગીના કોઈ પણ ત્રણ ગાણિતિક વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
3. તમારી પસંદગીના કોઈ પણ ત્રણ આંકડાકીય વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
4. તમારી પસંદગીના નિર્ણય લેવામાં ઉપયોગી કોઈ પણ ત્રણ વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
5. તમારી પસંદગીના કોઈ પણ ત્રણ તારીખ અને સમયને લગતા વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
6. તમારી પસંદગીના કોઈ પણ ત્રણ શાબ્દિક લખાણ ઉપરના વિધેય (સ્ટ્રિંગ ફંક્શન) યોગ્ય ઉદાહરણ સાથે સમજાવો.
7. કેલ્સીમાં ઉપલબ્ધ IF વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
8. ત્રણ તાર્કિક વિધેય યોગ્ય ઉદાહરણ સાથે સમજાવો.
9. નીચેનામાંથી સાચો વિકલ્પ જણાવો :
 - (1) નીચેનામાંથી કઈ રીતે આપણે કેલ્સીમાં વિધેય દાખલ કરી શકીએ ?
 - (a) સેલમાં વિધેયનું નામ સીધેસીધું ટાઈપ કરીને
 - (b) ફંક્શન વિઝાર્ડ અથવા ટૂલબાર ઉપરથી પસંદ કરીને
 - (c) (a) અને (b) બંને
 - (d) વિધેય આધારિત
 - (2) વિધેયની શરૂઆત નીચેનામાંથી કયા વિકલ્પ સાથે થાય છે ?

(a) '=' ચિહ્ન	(b) મૂળાક્ષરો	(c) સંખ્યા	(d) આમાંનો કોઈ પણ વિકલ્પ
---------------	---------------	------------	--------------------------
 - (3) નીચેનામાંથી કયો તાર્કિક વિધેય નથી ?

(a) OR	(b) AND	(c) NOT	(d) PROPER
--------	---------	---------	------------
 - (4) આપેલું વર્ષ લીપ યર છે કે નહીં તે શોધવા માટે તમે કેલ્સીના સમય અને તારીખને લગતાં વિધેયોમાંથી કયો વિધેય વાપરશો ?

(a) DATE	(b) TIMESTAMP	(c) YEARS	(d) YEARDIFF
----------	---------------	-----------	--------------

- (5) કેલ્સીના સેલમાં વિધેય દાખલ કરવા માટે નીચેનામાંથી કઈ યુક્તિ શક્ય છે ?
 (a) ફંક્શન વિઝાર્ડ દ્વારા (b) જાતે વિધેય દાખલ કરીને
 (c) (a) અને (b) બંને (d) વિધેય આધારિત
- (6) કેલ્સીનાં વિધેયમાં કેટલા ચલ વાપરી શકાય ?
 (a) એક (b) બે (c) ત્રણ (d) વિધેય આધારિત
- (7) વિધેયનો ચલ શું હોઈ શકે ?
 (a) કિંમત (b) શાબ્દિક લખાણ (c) અન્ય વિધેય (d) આપેલા બધા

પ્રાયોગિક સ્વાધ્યાય

- તમારા જુદા જુદા વિષયોના ગુણ લઈ એક સારું ગુણપત્રક બનાવો અને તેમાં શાળાનું નામ, વિદ્યાર્થીનો નંબર, વિદ્યાર્થીનું નામ, ધોરણ અને ગુણનો સમાવેશ કરો. આપેલા ગુણને આધારે ટકા અને વર્ગ (પ્રથમ વર્ગ, પાસ, નાપાસ વગેરે) પણ શોધો. આ કામ માટે તમે તમારું ગયા વર્ષનું નિશાળનું રિપોર્ટ કાર્ડ પણ વાપરી શકો.
- કેલ્સી વાપરીને તમારા ઘરનું વીજળીનું બિલ બનાવો.
- તમારા ઘરનાં સદસ્યને કે પડોશીને લોન બાબતની માહિતી પૂછો. NPER વિધેય વાપરીને લોન પરત કરવાનો સમયગાળો શોધો.
- ત્રણ જુદા જુદા સેલમાં ત્રણ સંખ્યાઓ લખો. આ કિંમતો એવી લખો કે જે એક માન્ય તારીખ બની રહે. હવે DATE વિધેય વાપરીને આ કિંમતોને માન્ય તારીખમાં ફેરવો.
- કોઈ તારીખ લઈ શોધી કાઢો કે તે લીપ યર છે કે નહીં ?
- મોબાઈલ, આઈ-ફોન અને આઈ-પેડ માટેનાં સ્પ્રેડશીટ પેકેજ વિશે માહિતી મેળવવા પ્રયત્ન કરો. આ જાતની સગવડ ધરાવતાં પેકેજ / સ્પ્રેડશીટનાં નામ શોધી કાઢો.



કેલ્સીમાં આલેખની રચના

ટેક્નોલોજીના વિકાસ અને તકનીકી વિશ્લેષણની વધતી જરૂરિયાતોને કારણે આલેખ(ચાર્ટ-chart)નો ઉપયોગ ઘણો વધ્યો છે. સ્પ્રેડશીટ પેકેજ વ્યાપકપણે પ્રચલિત થવાનું એક કારણ આલેખની ઉપયોગિતા છે. આલેખ એ વિશાળ માહિતીને આકર્ષક રીતે રજૂ કરવાની ક્ષમતા ધરાવતી વ્યાવસાયિક સ્થિતિનું એક ઉદાહરણ છે. આલેખને આંકડાકીય માહિતી ચિત્રાત્મક સ્વરૂપે રજૂ કરવાની રીત પણ કહેવામાં આવે છે.

આલેખ તૈયાર કરવા માટે કોઈ પણ વ્યક્તિને વિવિધ આલેખોનું મૂળભૂત જ્ઞાન અને આલેખનાં સ્વરૂપો તેમજ તેના વિનિયોગોના ખ્યાલ વિશે જાણકારી હોવી જોઈએ. એટલે કે, સ્પ્રેડશીટ પેકેજમાં માત્ર કોઈ પણ પ્રકારનો આલેખ પસંદ કરીને આલેખ તૈયાર કરી દેવાથી વ્યવસાયમાં તેનું ખાસ પ્રદાન રહેતું નથી. તમને આલેખનાં ખ્યાલોની સમજણ હોવી જ જોઈએ કે આલેખ તમારા વ્યવસાયમાં કઈ રીતે મદદરૂપ થશે. આ ઉપરાંત દરેક વ્યક્તિએ એ પણ જાણવું જોઈએ કે કોઈ ચોક્કસ પ્રકારનો આલેખ (ચાર્ટ) ક્યારે વાપરવો જોઈએ, તે કયા પ્રકારની માહિતી પૂરી પાડે છે, તે કયા પ્રકારનો આધાર પૂરો પાડે છે અને આ આલેખથી કયા પ્રકારના નિર્ણય લઈ શકાય છે.

આલેખના પ્રકારો (Type of charts)

વિનિયોગની પ્રકૃતિ અને આપણી જરૂરિયાત પ્રમાણે આલેખનો પ્રકાર પસંદ કરવો જોઈએ. આલેખ વિવિધ કાર્યો માટે વપરાય છે જેમ કે : ઇતિહાસ કહેવા માટે, વિકલ્પોના મૂલ્યાંકન માટે, કોઈ વલણ રજૂ કરવા માટે અને અસાધારણ કિસ્સાઓ શોધવા માટે. આનો અર્થ એ થાય કે ખોટા પ્રકારના આલેખની પસંદગી ખ્યાલોની અપૂરતી રજૂઆત કરે છે અને ક્યારેક ગેરસમજ ઉત્પન્ન કરે છે. જ્યારે બીજી બાજુએ, યોગ્ય પ્રકારના આલેખની પસંદગી સાચા અને ઝડપી નિર્ણયો લેવામાં મદદરૂપ થાય છે. આપણે શા માટે આલેખ વાપરવા જોઈએ તેનાં પ્રચલિત કારણો નીચે મુજબ છે:

- સરખામણી માટે
- વિતરણની સમજૂતી આપવા માટે
- સ્થિતિ સમજવા માટે
- અમુક ચોક્કસ સમયગાળાના વલણનું વિશ્લેષણ કરવા માટે
- નિયત પરિણામમાં ફેરફારની બારીક તપાસ કરવા માટે
- વાસ્તવિક વસ્તુઓ (એન્ટિટી) વચ્ચેના સંબંધો ઓળખવા અને સમજવા માટે

કોઈપણ પરિસ્થિતિમાં આલેખ તૈયાર કરવા માટે નીચે આપેલાં પગલાંને અનુસરો :

- તમે જે કંઈ કહેવા ઇચ્છતા હોય તે બાબત સ્પષ્ટ બનો.
- માહિતી (ડેટા) ભેગી કરો અને ગોઠવો.
- અમાન્ય ડેટા કાઢી નાખો.
- સૌથી વધુ યોગ્ય આલેખનો પ્રકાર નક્કી કરો અને ખાતરી કરો કે આલેખ માટે એકઠો કરેલો ડેટા પૂરતો છે કે કેમ.
- આલેખ (ચાર્ટ) બનાવો અને
- ચાર્ટને વધારે આકર્ષક અને યોગ્ય બનાવો (ફોર્મેટ કરો).

ઉપર જણાવ્યા પ્રમાણે ધ્યેય નક્કી કર્યા પછી આલેખ બનાવવા માટે સૌ પ્રથમ પગલું ડેટા તૈયાર કરવાનું છે. એક વાર સ્પ્રેડશીટના દસ્તાવેજમાં ડેટા દાખલ કરી દેવામાં આવે પછી તમે તેને ચિત્રાત્મક (ગ્રાફિકલ) સ્વરૂપમાં જોઈ શકો છો.

આલેખ (ચાર્ટ) ઉમેરવો (Inserting a chart)

વર્કશીટમાં આલેખ ઉમેરવા માટે નીચે પ્રમાણે કાર્ય કરો :

- સ્પ્રેડશીટમાં ડેટાનો વિસ્તાર પસંદ કરો.
- હવે પસંદ કરો : **Insert → Chart**

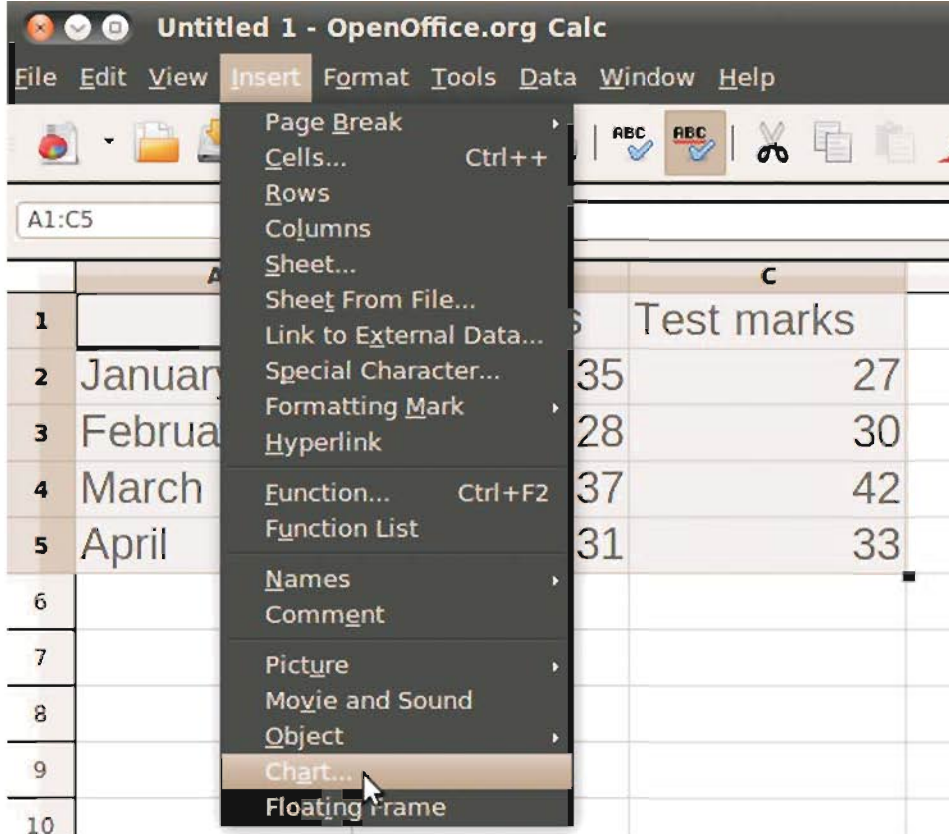
ધારો કે તમને શૈક્ષણિક વર્ષમાં જાન્યુઆરીથી એપ્રિલ દરમિયાન લીધેલી અણધારી કસોટી (surprise quizzes) અને વર્ગ પરીક્ષાઓના ગુણ કોષ્ટક 8.1માં જણાવ્યા પ્રમાણે આપ્યા છે.

	Quiz marks	Test marks
January	35	27
February	28	30
March	37	42
April	31	33

કોષ્ટક 8.1 : વિદ્યાર્થીના ગુણ

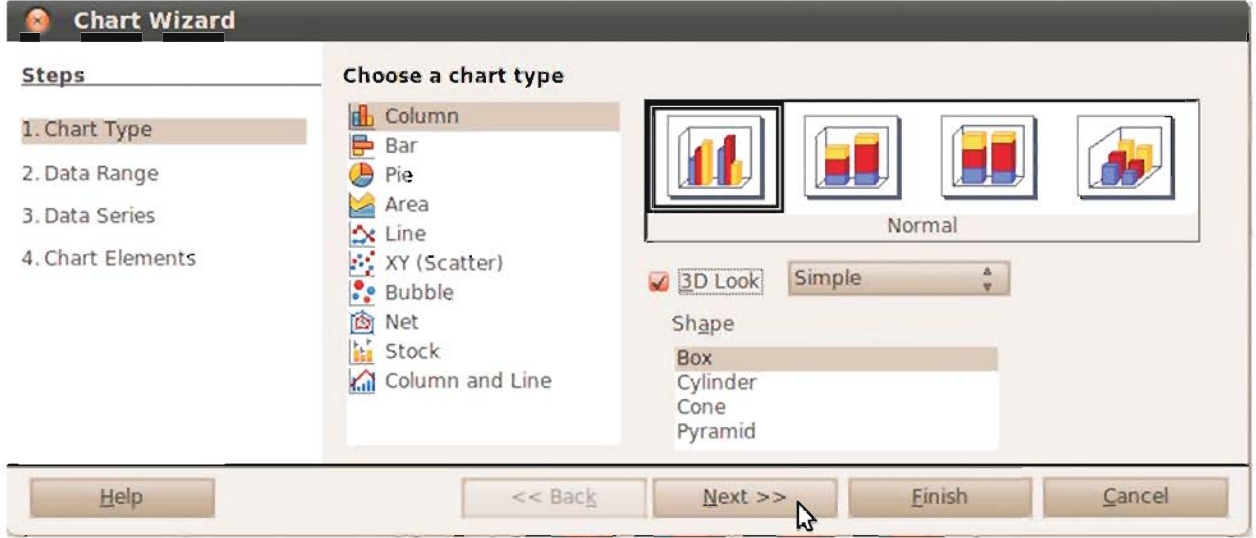
સ્પ્રેડશીટ દસ્તાવેજ ખોલો અને કોષ્ટક 8.1માં દર્શાવ્યા પ્રમાણે ડેટા દાખલ કરો. ડેટાનો ક્રમ બદલશો નહીં અન્યથા જ્યારે તમે આલેખ બનાવશો ત્યારે તે અલગ પરિણામ આપશે. ડેટાનો સંગ્રહ પણ વારંવાર કરો.

એકવાર તમે સંપૂર્ણ ડેટા દાખલ કરી દો પછી ડેટાનો વિસ્તાર પસંદ કરો. અહીં આપણા કિસ્સામાં ડેટાનો વિસ્તાર સેલ A1 થી સેલ C5 છે. ડેટાનો વિસ્તાર પસંદ કરેલો રાખીને જ હવે **Insert** વિકલ્પ આપો. એક ઊભું સબમેનૂ સ્ક્રીન ઉપર દેખાશે. તેમાંથી **Chart** વિકલ્પ પસંદ કરો. કાર્યની શ્રેણી આકૃતિ 8.1માં દર્શાવી છે.



આકૃતિ 8.1 : આલેખ ઉમેરવો

જ્યારે તમે આલેખ (ચાર્ટ) ઉમેરવાનો વિકલ્પ પસંદ કરો છો ત્યારે તમે ચાર્ટ વિઝાર્ડ (chart wizard) જોઈ શકશો. કેલ્સીમાં ચાર્ટ વિઝાર્ડ એ હકીકતમાં ઓડેશીટ પેકેજમાં ચાર્ટ તૈયાર કરવાની ક્રિયાઓની એક સરળ અને ઉપયોગકર્તાઓ સાથે મૈત્રીસભર પગલાંની એક શ્રેણી છે. કોઈ પણ વ્યક્તિએ માત્ર ડેટાનો વિસ્તાર પસંદ કરવાનો હોય છે અને તે પછી ચાર્ટ વિઝાર્ડ ચાલુ (ઇન્વોક) કરવાનું હોય છે. બાકીનાં પગલાંનું ધ્યાન વિઝાર્ડ લઈ લેશે. ચાર્ટ વિઝાર્ડનું ડાયલોગ બોક્સ આકૃતિ 8.2માં દર્શાવ્યા પ્રમાણેનું હોય છે.



આકૃતિ 8.2 : ચાર્ટ વિઝાર્ડ (Chart wizard)

આકૃતિ 8.2માં દર્શાવેલું ચાર્ટ વિઝાર્ડ ચાર પગલાં જણાવી ચાર્ટનો પ્રકાર (chart type), વિગતોનો વિસ્તાર (data range), વિગતોની શ્રેણી (data series) અને ચાર્ટના ઘટકો (chart elements)ની માહિતી લે છે. આ ચાર પગલાંની ચર્ચા નીચે કરેલી છે :

ચાર્ટનો પ્રકાર પસંદ કરવો (Selecting chart type)

આકૃતિ 8.2માં દર્શાવ્યા પ્રમાણે column આલેખ પસંદ કરો. તે આલેખ કેવો દેખાશે તે પણ આપણે અગાઉથી જોઈ શકીએ છીએ. દરેક સમયે આપણે આલેખનો પ્રકાર બદલીએ તે સાથે તે આલેખ કેવો દેખાશે (preview) તે પણ બદલાશે, જે આપણને આલેખ કેવો દેખાશે તેની સુંદર સમજ આપે છે.

ચાર્ટ વિઝાર્ડના મુખ્ય ભાગ નીચે પ્રમાણે છે :

- આલેખ તૈયાર કરવા અનિવાર્ય બધાં પગલાંની યાદી
- આલેખના પ્રકારોની યાદી
- વિવિધ પ્રકારના આલેખ અગાઉથી જોવાની સગવડ (preview)
- દરેક પ્રકારના આલેખના વિકલ્પો અને
- આગળ વધવા માટે અથવા પાછા જઈને બદલવાના વિકલ્પો

ચાર્ટ વિઝાર્ડમાં તમે અન્ય પ્રકારના આલેખ અને તેના પ્રિવ્યૂ જોઈ શકો છો. આલેખ(ચાર્ટ)ના પ્રકારોમાં કોલમ ચાર્ટ (Column chart), બાર ચાર્ટ (Bar chart), પાઈ ચાર્ટ (Pie chart), લાઈન ચાર્ટ (Line chart), એરિયા ચાર્ટ (Area chart), XY ચાર્ટ, બબલ ચાર્ટ (Bubble chart), નેટ ચાર્ટ (Net chart), સ્ટોક ચાર્ટ (Stock chart), કોલમ એન્ડ લાઈન ચાર્ટ (Column and line chart)નો સમાવેશ થાય છે. કેલ્સી 10 પ્રકારના મૂળભૂત આલેખની પસંદગી આપણને આપે છે. એક વખત chart type આપણે પસંદ કરીએ પછી તેના અન્ય વિકલ્પો જેવા કે આલેખનો દેખાવ અને આકાર પસંદ કરી શકો છો. કેલ્સી દરેક પ્રકારના આલેખ માટે થોડા વિકલ્પો પૂરા પાડે છે. તમે જે આલેખની પસંદગી કરો તે પ્રમાણે તેના વિકલ્પો બદલાશે. કોઈ પણ તબક્કે તમે પાછા અગાઉનાં પગલાં ઉપર જઈ શકો અને તમે પસંદ કરેલા વિકલ્પો બદલી શકો છો.

અહીં 3D view box (લંબચોરસ જગ્યા) પણ હોય છે. જો તમે 3D boxને (તેના ઉપર ક્લિક કરીને) પસંદ કરશો તો તમે બદલાયેલ પ્રિવ્યૂ જોઈ શકશો. ફક્ત 3D ને ચોક્કસ આલેખ જ (જેવા કે કોલમ ચાર્ટ, બાર ચાર્ટ, પાઈ ચાર્ટ અને એરિયા ચાર્ટ) 3Dમાં જોવાનો વિકલ્પ આપે છે. તમે આલેખનો આકાર Shape boxમાંથી "Box", "Cylinder", "Cone" અને "Pyramid" પસંદ કરી શકો છો. તમે જે પસંદ કરો તે પ્રમાણે ચોક્કસ પ્રિવ્યૂ તમને જોવા મળશે.

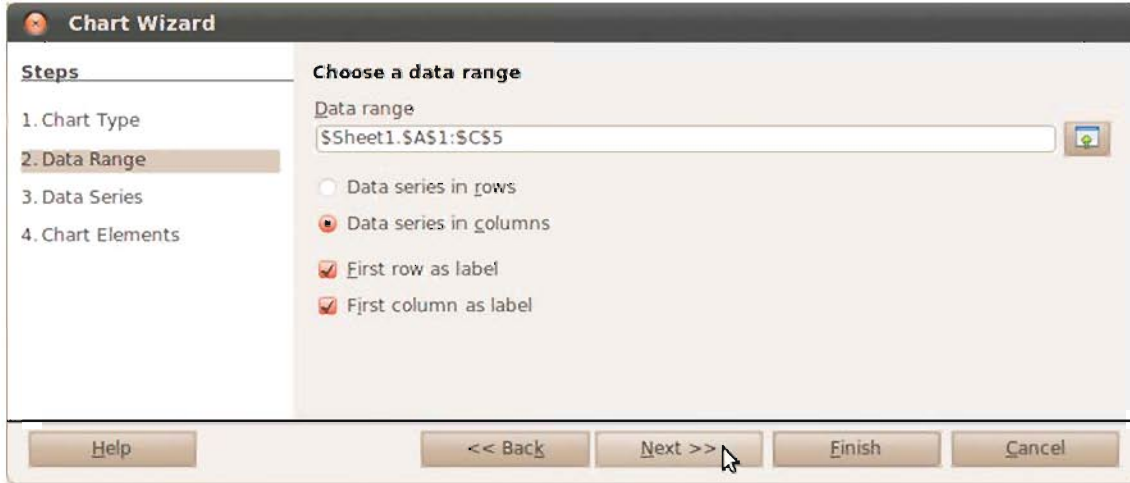
ઉદાહરણ તરીકે, આપણે નીચે જણાવેલા વિકલ્પો પસંદ કરેલા છે :

- ચાર્ટ ટાઈપ - Column
- ચાર્ટ લોક - 3D સાથે Simple
- ચાર્ટ શેઈપ - Box

અંતમાં, Next બટન દબાવો. તમે નિરીક્ષણ કરી શકશો કે કેલ્સીએ ચાર્ટ વિઝાર્ડની પશ્ચાદ્ભૂમિમાં એક આલેખ બનાવી દીધો છે. આ તબક્કે, જો તમે Finish બટન દબાવશો તો આ જ ડિફોલ્ટ ચાર્ટ તમે જોઈ શકશો કે જે તમે પછી તેને સુધારી કે ફોર્મેટ કરી શકો છો. તમે અગાઉના તબક્કામાં જવા માટે Back બટન દબાવી શકો અને તમારા વિકલ્પોમાં ફેરફાર પણ કરી શકો છો.

ડેટા વિસ્તાર (Data range)

એક વખત તમે આલેખનો પ્રકાર નક્કી કરી દો પછી તમારે આલેખ માટે જરૂરી ડેટા આપવા પડે. એક શક્યતા એ છે કે આપણે અહીં ક્યું છે તેમ Insert → Chart આપતાં પહેલાં જ ડેટાનો વિસ્તાર આપી દીધો હોય. જો તેમ ન ક્યું હોય તો આકૃતિ 8.3માં દર્શાવ્યા પ્રમાણે આપણે હજી Data range વિકલ્પ પસંદ કરીને આપી શકીએ છીએ.



આકૃતિ 8.3 : Data rangeની પસંદગી

જો આલેખ ઉમેરતાં પહેલાં આપણે સંપૂર્ણ ડેટા વિસ્તારને બદલે ડેટા વિસ્તારનો અંશ જ પસંદ કરેલો હોય તો કેલ્સી ડેટા વિસ્તાર ઓળખવા પૂરતું ચકોર છે. એટલે કે, ધારો કે તમે ફક્ત એક જ સેલ જેમ કે B3 પસંદ કરેલો છે અને તમે Insert → Chart પસંદ કરો છો તો આ સંજોગોમાં કેલ્સી પોતાની જાતે ડેટાનો વિસ્તાર ધારી લેશે. જો કે ડેટા વિસ્તાર ચકાસી લેવી એ એક સારી પદ્ધતિ છે. તમે અન્ય સ્પ્રેડશીટ દસ્તાવેજના હયાત ડેટા અથવા અન્ય સુસંગત (કમ્પેટિબલ) દસ્તાવેજના ડેટાનો પણ નીચે જણાવ્યા પ્રમાણે ઉપયોગ કરી શકો છો.

કેલ્સીના આલેખ નીચે જણાવેલા ડેટા આધારિત હોય છે :

- કેલ્સી સેલના વિસ્તારમાંથી સ્પ્રેડશીટની કિંમતો
- Writer ટેબલમાંથી સેલની કિંમતો. તમે Writer, Draw અથવા Impress જેવા પેકેજમાં પણ ડેટા દાખલ કરીને પછી તેની નકલ (copy and paste) કેલ્સી સ્પ્રેડશીટમાં કરી શકો છો.
- કોઈ પણ યોગ્ય રીતે ગોઠવેલી શાબ્દિક માહિતીની ફાઈલ (ફોર્મેટેડ ફાઈલ) અથવા ટેબલ.

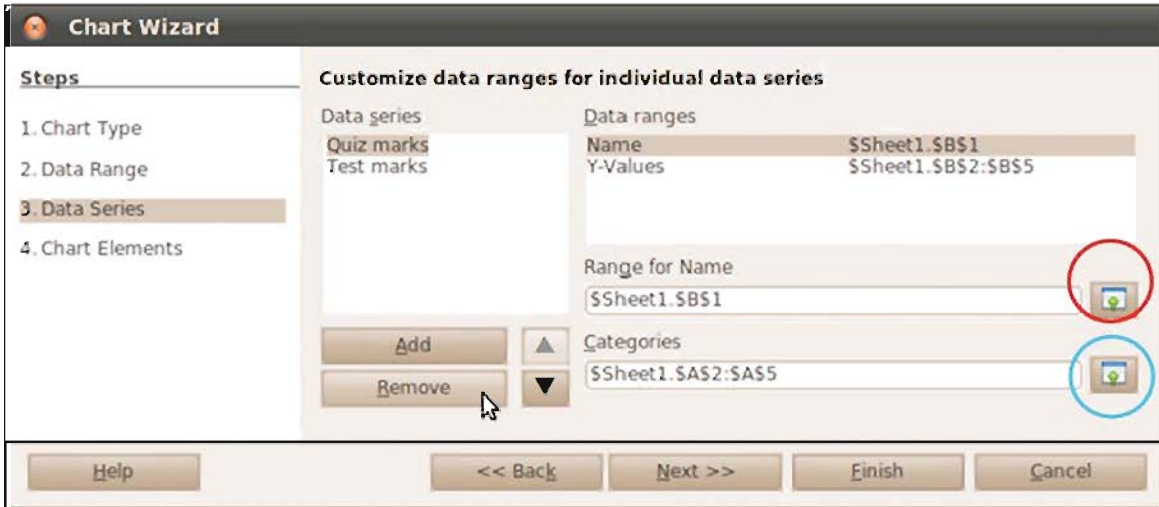
આપણા ઉદાહરણમાં ડેટા આ જ શીટમાંથી આવે છે અને તેની વિસ્તાર A1 થી C5 છે. અહીં નોંધ કરો કે કેલ્સીએ ડેટાનો વિસ્તાર \$Sheet1.\$A\$1:\$C\$5 લીધો છે. એટલે કે, કેલ્સી વિસ્તારને નિરપેક્ષ સ્થાનાંકમાં ફેરવી દે છે. આથી, જો આલેખને અન્ય કોઈ સ્થાન ઉપર લઈ જવામાં આવે તો પણ ડેટાના વિસ્તારમાં કોઈ ફેરફાર ન થાય (અચલ). તે પછીનું બટન રો અથવા કોલમમાં ડેટા શ્રેણીનું (data series) સ્થાન પૂછે છે. અહીં આપણે આપણા ગુણ કોલમ

(ઊભી હરોળ)માં દાખલ કરેલા છે. આથી આપણે **Data series in column** વિકલ્પ પસંદ કરી શકીએ. આ ઉપરાંત આપણે **First row as label** વિકલ્પ પણ પસંદ કરવો પડશે. જ્યારે ડેટા અનેક ઊભી હરોળોમાં (કોલમ્સ) હોય અને દરેકમાં શીર્ષક હોય ત્યારે આ વિકલ્પ ઘણો ઉપયોગી બને છે. પહેલી રો (આડી હરોળ) દરેક કોલમ(ઊભી હરોળ)નાં બધાં શીર્ષક બને છે. આપણે પહેલી હરોળ બધાં શીર્ષક ધરાવે છે તે જણાવીને આપણે કોલમનાં શીર્ષક આપી શકીએ છીએ. વિદ્યાર્થીના ગુણનાં ઉદાહરણમાં પહેલી આડી હરોળ **Quiz Marks** અને **Test Marks** જેવાં શીર્ષકો ધરાવે છે.

આ જ રીતે, જ્યારે દરેક આડી હરોળમાં શીર્ષક હોય તો આપણે **First column as label** વિકલ્પ પસંદ કરી શકીએ. તમે બંને ચેકબોક્સ (check boxes) પણ પસંદ કરી શકો.

ડેટા શ્રેણી (Data series)

ચાર્ટ વિઝાર્ડ આલેખ તૈયાર કરવા માટે તમારી અનુકૂળતા પ્રમાણે ડેટાનો વિસ્તાર બનાવવાની સગવડ પૂરી પાડે છે. અહીં, આપણે ચાર્ટ વિઝાર્ડ દ્વારા જાતે જ પસંદ કરેલી કિંમતોનો ઉપયોગ કરીશું. આકૃતિ 8.4માં ચાર્ટ વિઝાર્ડ દ્વારા પસંદ કરેલી ડેટા શ્રેણી દર્શાવેલી છે.



આકૃતિ 8.4 : ડેટા શ્રેણીની પસંદગી

તમે આ ડાયલોગ બોક્સ વડે ડેટા શ્રેણી ઉમેરી કે રદ કરી શકો છો. જેમ કે આપણા ઉદાહરણમાં અંતિમ પરીક્ષાના ગુણ ઉમેરવા માટે **Add** બટનનો ઉપયોગ કરીને ડેટાની શ્રેણી ઉમેરી શકાય છે. કોઈ ડેટાની શ્રેણી રદ કરવા માટે પ્રથમ શ્રેણી પસંદ કરો કે જે તમે રદ કરવા ઇચ્છા છો તે પછી **Remove** (આકૃતિ 8.4માં જુઓ માઉસનું તીર) પસંદ કરો. ડાયલોગ બોક્સ વિસ્તારનું નામ અને વિસ્તારની કિંમતો આપવાની સગવડ પણ પૂરી પાડે છે. અહીં વિસ્તારનું નામ **\$B\$1** છે અને તેની કિંમત "Quiz marks" છે. તમે **Select** બટન (આકૃતિમાં લાલ લીટી સાથે વર્ણુલ દોરેલું છે) પણ વાપરી શકો છો. તે ચાર્ટ વિઝાર્ડને નાનું બનાવી દેશે અને તમને વર્કશીટના વિસ્તારમાં લઈ જશે. તમે તમારી જરૂરિયાત પ્રમાણે યોગ્ય સેલ પસંદ કરી શકો છો. જ્યારે તમે પસંદગીનું કાર્ય પૂરું કરો તે પછી ચાર્ટ વિઝાર્ડનું **Maximize** બટન દબાવો. જ્યારે તમને સેલનાં સ્થાનાંક યાદ ન હોય ત્યારે આ ઉપયોગી બને છે.

આ જ પ્રમાણે તમે ડેટા શ્રેણીની કિંમતો પસંદ કરી શકો છો. આપણા ઉદાહરણમાં ડેટા શ્રેણીની કિંમતો સેલ B2થી B5માં રહેલા quiz marks છે. તમે **\$Sheet1.\$B\$2: \$B\$5** પણ લખી શકો છો.

Data seriesના તબક્કામાં તમે ડેટાની categories પણ દાખલ કરી શકો છો. આપણા ઉદાહરણમાં Data series Categories સેલ A2થી A5માં રહેલા મહિનાનાં નામ છે. તમે **\$Sheet1.\$A\$2: \$A\$5** પણ લખી શકો છો. અહીં પણ તમે **Select** બટન (આકૃતિ 8.4માં વાદળી લીટીથી સાથે ગોળ નિશાની કરેલું) વડે ચાર્ટ વિઝાર્ડને નાનું કરી શકો અને સ્પ્રેડશીટમાંથી કિંમતો સીધી જ પસંદ કરી શકો છો. અંતમાં **Maximize** બટન દબાવો.

ડેટા એલિમેન્ટ્સ (Data elements)

ચાર્ટ વિઝાર્ડ આલેખના જુદા જુદા ભાગનાં નામ જેમ કે Title, Subtitle, X-axis captions, Legendનું સ્થાન અને Grids પ્રદર્શિત કરવાના વિવિધ વિકલ્પો દાખલ કરવાની સગવડ આપે છે. આકૃતિ 8.5માં ચાર્ટ વિઝાર્ડનો સ્ક્રીન આપેલો છે કે જેમાં ડેટા એલિમેન્ટ્સ સેટ કરવાની સગવડ મળે છે.

Chart Wizard

Steps

1. Chart Type
2. Data Range
3. Data Series
4. Chart Elements

Choose titles, legend, and grid settings

Title: My Performance

Subtitle: Year 2013

X axis: Month

Y axis: Marks

Z axis:

Display grids: ☒ X axis ☒ Y axis ☐ Z axis

☒ Display legend

☐ Left

☒ Right

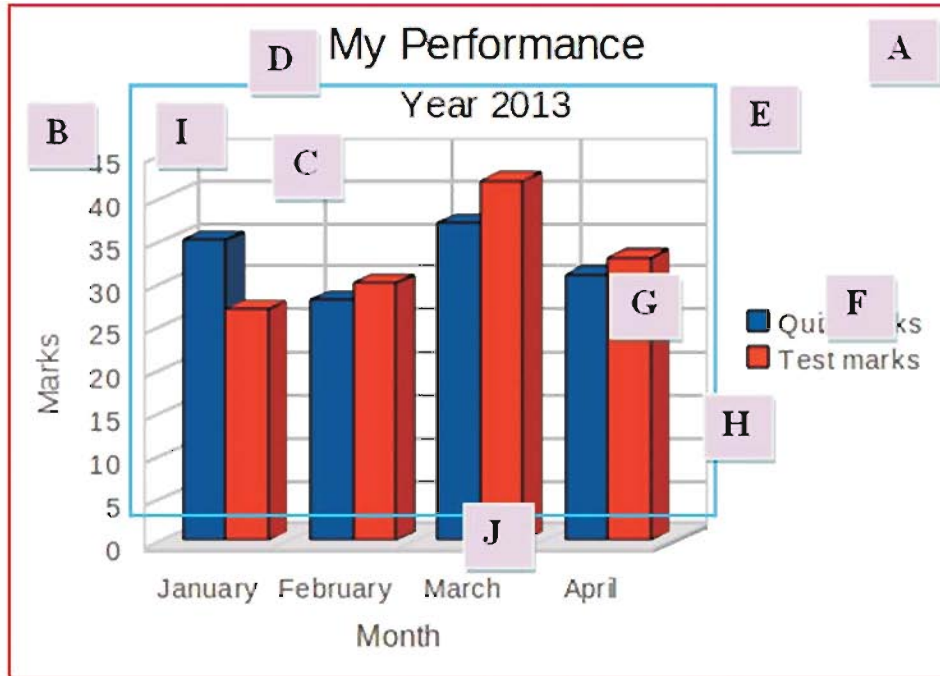
☐ Top

☐ Bottom

Help << Back Next >> Finish Cancel

આકૃતિ 8.5 : ચાર્ટના ઘટકો

ચાર્ટ વિઝાર્ડના આ તબક્કામાં ચાર્ટ વિશેની અલગ અલગ માહિતી જેવી કે આલેખનું શીર્ષક, આલેખનું પેટા શીર્ષક અને અક્ષ તેમજ લિજેન્ડની માહિતી પૂરી પાડવામાં આવે છે. આકૃતિ 8.5માં દર્શાવ્યા પ્રમાણે **Chart elements**નો પહેલો વિકલ્પ **Title** છે. અહીં તમે આલેખનું મુખ્ય શીર્ષક આપી શકો છો. બીજો વિકલ્પ છે **Subtitle**, જ્યાં તમે આલેખનું પેટા શીર્ષક આપી શકો. શીર્ષક અને પેટા શીર્ષક આલેખ વિસ્તારના સૌથી ઉપરના ભાગમાં રાખવામાં આવે છે. તમે શીર્ષક અને પેટા શીર્ષક આપી દો તે પછી તમે અક્ષ વિશે માહિતી આપી શકો. ત્રણ અક્ષ X-અક્ષ, Y-અક્ષ અને Z-અક્ષનાં શીર્ષકની માહિતી અહીં આપી શકો છો. આ તબક્કાએ તમારે **Display legend** વિકલ્પ વડે લિજેન્ડ પ્રદર્શિત કરવા છે કે નહીં તે પણ પસંદ કરી શકો છો. જો તમે પ્રદર્શિત કરવાનું પસંદ કરો તો તે લિજેન્ડ તમારે આલેખની ડાબી, જમણી, ઉપર કે નીચે ક્યાં પ્રદર્શિત કરવા ઇચ્છો છો તે જણાવવું પડશે. જો કોઈ વિકલ્પ ન આપો તો પૂર્વનિર્ધારિત રીતે આલેખ વિસ્તારમાં આલેખની જમણી બાજુએ લિજેન્ડ દેખાશે.



આકૃતિ 8.6 : સાદો આલેખ અને ચાર્ટ એલિમેન્ટ્સ

બધા વિકલ્પો સાચા આપી દીધા પછી તૈયાર આલેખ માટે તમે Finish બટન દબાવો. હવે ચાલુ શીટમાં તમને આલેખ દેખાશે. તમે આકૃતિ 8.6માં દર્શાવ્યા પ્રમાણે ચાર્ટ ઘટકોનાં લેબલ સાથે આલેખ જોઈ શકશો. આલેખના વિવિધ ભાગની ચર્ચા હવે પછીના વિભાગમાં કરી છે.

આલેખના વિવિધ ઘટકો (Elements of a chart)

આલેખમાં સામાન્ય રીતે આ મુજબના ઘટકો હોય છે : આલેખ વિસ્તાર, આલેખનું શીર્ષક અને પેટા શીર્ષક, કિંમતો સાથે આલેખના અક્ષ, આલેખ અક્ષ ડેટા વિસ્તાર, આલેખ ડેટા વિસ્તાર, આલેખની અક્ષનાં લિજેન્ડ, ડેટા શ્રેણી અને આલેખ જગ્યા (ચાર્ટ ફ્લોર). આપણે અત્યાર સુધીમાં આકસ્મિક કસોટી અને વર્ગ પરીક્ષાઓમાં વિદ્યાર્થીના દેખાવ બાબતનો એક સાદો આલેખ બનાવી દીધો છે. આલેખના કોઈ ચોક્કસ ભાગ વિશે જાણકારી મેળવવા માટે તમે ફક્ત માઉસના તીરને તે જગ્યાએ મૂકો. આથી તે ભાગનું નામ પ્રદર્શિત થશે. ચાલો, આપણે કેલ્સીના આલેખનાં કેટલાક ઘટકોનું નિરીક્ષણ કરીએ. આકૃતિ 8.6માં કેટલાંક લેબલ આલેખના જુદા જુદા ભાગનો નિર્દેશ કરે છે. આ લેબલની સમજૂતી નીચે આપેલી છે:

લેબલ A : આલેખ વિસ્તાર (Chart area) : આલેખનો વિસ્તાર લંબચોરસ વિસ્તાર છે જેમાં આલેખનો સમાવેશ થયેલો છે.

લેબલ B : કિંમતો સાથે Y-અક્ષ (Y-axis with values) : આપણે જે ડેટા આપેલો હોય તેના આધારે Y-અક્ષની કિંમતો જાતે જ લેવામાં આવે છે. અહીં આપણે 50થી વધારે ગુણ આપેલા નથી આથી Y-અક્ષ ઉપર 0 થી 45 કિંમત લેવામાં આવી છે.

લેબલ C : અક્ષ ડેટા વિસ્તાર (Data area axis) : અક્ષ ડેટા વિસ્તાર એ ડેટા વિસ્તારની અંદર અક્ષ છે. આ વિકલ્પ પસંદ કરવાથી આલેખની પશ્ચાદ્ભૂમિમાં તમે આડી અને ઊભી રેખાઓનું જાળું જોઈ શકો છો. ડેટા વિસ્તારની અંદર અક્ષનો મુખ્ય ઉપયોગ કિંમતો જાણવાનો છે જ્યારે આલેખમાં કિંમતો પ્રદર્શિત ન કરી હોય.

લેબલ D : શીર્ષક અને પેટા શીર્ષક (Title and Subtitle) : આખા આલેખને આપેલું કામ એ આલેખનું શીર્ષક છે. અહીં આપણે My Performance શીર્ષક આપેલું છે. શીર્ષક સાથે આપણે પેટા શીર્ષક પણ આપી શકીએ. પેટા શીર્ષક આપવાનો હેતુ વધારાની માહિતી અથવા આલેખનો ઉપયોગ બતાવવાનો હોય છે. અહીં આપણે Year 2013 પેટા શીર્ષક તરીકે આપેલું છે.

લેબલ E : ડેટા વિસ્તાર (Data area) : આલેખ વિસ્તારની અંદર ડેટા વિસ્તારમાં આલેખ સમાયેલો હોય છે. જે આલેખ તરીકે પણ ઓળખાય છે. આલેખ વિસ્તારમાં રહેલો આલેખ એ મુખ્ય ભાગ છે અને આથી આલેખ વિસ્તારની મધ્યમાં તે રાખવામાં આવે છે. આકૃતિ 8.6માં વાદળી રંગની સીમારેખાથી ઘેરાયેલો વિસ્તાર આલેખ (chart) અથવા ડેટા વિસ્તાર (data area) તરીકે ઓળખાય છે.

લેબલ F : લિજેન્ડ (Legend) : આલેખ એક પ્રકારનું ચિત્રકામ છે. તેમાં ડેટા અને ગ્રાફિક્સને સંબંધિત કેટલીક શાબ્દિક માહિતી હોવી જોઈએ. આકૃતિ 8.6માં ઊભી હરોળો (કોલમ) વાદળી અને લાલ રંગમાં દર્શાવેલી છે. તમે શીર્ષક, X-અક્ષ અને Y-અક્ષના લેબલ ઉપરથી જાણી શકો કે આ આલેખ જુદા જુદા માસમાં લીધેલી વિવિધ કસોટીઓના ગુણનો છે. જો કે, અહીં એ સ્પષ્ટ નથી કે કયા રંગની કોલમ કયા ગુણનો નિર્દેશ કરે છે. આ માટેની માહિતીનો સમાવેશ કરવા માટે આલેખમાં લિજેન્ડનો ઉપયોગ થાય છે. અહીં આપણે આકૃતિ 8.6માં વાદળી રંગ આકસ્મિક કસોટીનાં ગુણ અને લાલ રંગ વર્ગ કસોટીના ગુણ દર્શાવવા માટે વાપરેલો છે.

લેબલ G : ડેટા શ્રેણી કોલમ (Data series column) : આલેખ એક કરતાં વધારે ડેટા શ્રેણીનો હોઈ શકે. દરેક ડેટા શ્રેણી માટે અલગ અલગ રંગ પસંદ કરવામાં આવે છે અને પછી આલેખ તૈયાર થાય છે. ઉદાહરણ તરીકે, કોલમ ચાર્ટ(column chart)માં દરેક ડેટા શ્રેણી માટે રંગની પસંદગી કરી તે રંગનાં અનેક સ્તંભ જુદા જુદા સમયગાળા કે તબક્કા માટે બનાવવામાં આવે છે. આપણા ઉદાહરણમાં આકસ્મિક કસોટીના ગુણની એક ડેટા શ્રેણી બનાવેલી છે અને વાદળી રંગ આપોઆપ પસંદ થયેલો છે. કેલ્સી દરેક માસના ગુણ દર્શાવીને વાદળી રંગના અનેક સ્તંભ બનાવે છે. આકૃતિ 8.6માં વાદળી રંગના સ્તંભ હકીકતમાં આકસ્મિક કસોટીના ગુણની ડેટા શ્રેણી દર્શાવે છે. આ જ પ્રમાણે, આકૃતિ 8.6માં લાલ રંગના સ્તંભ હકીકતમાં વર્ગપરીક્ષાના ગુણ દર્શાવે છે.

લેબલ H : ચાર્ટ ફ્લોર (Chart floor) : ચાર્ટ ફ્લોર એ એક ક્ષિતિજ સમાંતર સપાટી છે જે આલેખના વિવિધ ભાગનો પાથો છે. કોઈ ચોક્કસ પ્રકારના આલેખ જેમ કે બાર અને કોલમ પ્રકારના આલેખમાં તે ખૂબ ઉપયોગી છે. તમે આકૃતિ 8.6માં ચાર્ટ ફ્લોરને રાખોડી રંગથી ભરેલો જોઈ શકશો.

લેબલ I : ચાર્ટ વોલ (Chart wall) : ચાર્ટ વોલ એ ધરતીને કાટખૂણે રહેલી સપાટી છે જે આલેખના Y-અક્ષને ટેકો પૂરો પાડે છે. આકૃતિ 8.6માં દર્શાવ્યા પ્રમાણે તેમાં કોઈ રંગ ભરેલો નથી. જો કે તેમાં તમે તમારી પસંદગીનો રંગ ભરીને તે બદલી શકો છો.

લેબલ J : X-અક્ષ ઉપરનાં લિજેન્ડ (legends on X-axis) : તમને આપેલા એક સાદા આલેખને લો. આલેખના X-અક્ષ ઉપર ફક્ત કિંમતો જ છે. આકૃતિ 8.6માં દર્શાવ્યા પ્રમાણે X-અક્ષ ઉપર "January", "February" વગેરે જેવા મહિનાનાં નામનો ઉપયોગ આપણે કર્યો છે. આ નામ જાતે જ ખુલાસો કે સ્પષ્ટીકરણ કરી દે છે. કોઈ પણ વ્યક્તિ વગર મહેનતે જાણી શકે કે આપણે વર્ષના મહિનાઓ વિશે વાત કરીએ છીએ. જો આપણે વાહનની ઝડપ બાબત જણાવવા ઇચ્છતા હોઈએ તો ? અલબત્ત, તે માટે આપણે 10, 20, 30 વગેરે લખી શકીએ. પણ એવો કોઈ રસ્તો છે કે આ કિમી / કલાક અથવા મીટર / મિનિટ છે તે જણાવી શકે ? X-અક્ષ ઉપરના આ એકમો અને વસ્તુની શ્રેણી X-અક્ષ ઉપર લિજેન્ડ દ્વારા જણાવી શકાય છે. આ જ પ્રમાણે આપણે Y-અક્ષ ઉપર પણ લિજેન્ડ જણાવી શકીએ છીએ.

આલેખ તૈયાર કરવાનો અન્ય રસ્તો (An alternative way to create a chart)

ઉપર જણાવ્યા તે સિવાય અન્ય વિકલ્પરૂપે તમે સ્ટાન્ડર્ડ ટૂલબાર ઉપરથી ચાર્ટ (chart) આઈકોન પસંદ કરી શકો. જો તમે સ્ક્રીન ઉપર સ્ટાન્ડર્ડ ટૂલબાર જોઈ શકતા ન હો તો આપણે અગાઉ ચર્ચા કરી હતી તે પ્રમાણે View → Toolbars વડે તે ટૂલબારને દર્શ્યમાન કરી શકો છો. આકૃતિ 8.7માં ટૂલબાર ઉપર ચાર્ટ આઈકોન દર્શાવ્યો છે.



આકૃતિ 8.7 : ચાર્ટ આઈકોન

તમારે ફક્ત ત્રણ કાર્ય કરવાં પડશે :

1. સૌપ્રથમ શીટમાં ડેટા દાખલ કરો.
2. ડેટા પસંદ કરો અને ઉપર જણાવ્યા પ્રમાણે ચાર્ટ આઈકોન દબાવો.
3. હવે **Chart type** પસંદ કરો. તે પછી **Finish** દબાવો. તમે આ તબક્કે અન્ય વિકલ્પો ન આપો. જ્યારે તમે Finish બટન દબાવો છો, તે સાથે તમારો આલેખ તૈયાર થઈને વર્તમાન શીટ ઉપર જ દેખાશે. તમારા આલેખમાં બધા શીર્ષક અને લિજેન્ડ ન પણ હોઈ શકે. જો કે તમે ઇચ્છો તો તેને ફોર્મેટ કરીને વધારે આકર્ષક અને અસરકારક હંમેશાં બનાવી શકો છો. નીચેના વિભાગમાં ફોર્મેટ કરવાના અને આલેખને બદલવાના રસ્તા સમજાવ્યા છે.

આલેખને ફોર્મેટ કરવો અને બદલવો (Formatting and Modifying Charts)

અનેક સમયે તમારી જરૂરિયાત પ્રમાણે આલેખ તૈયાર કરવાની જરૂર પડે છે. એટલે કે, આલેખને તમારી શૈલી પ્રમાણે તમે બદલવા ઇચ્છો. ચાલો, આપણે આલેખને ફોર્મેટ (format) કરવાના વિવિધ વિકલ્પોની ચર્ચા કરીએ.

આલેખની જગ્યા અને કદ બદલવા (Modifying the place and size of the chart)

જ્યારે આલેખ બનાવવામાં આવે છે ત્યારે તેનો ચોક્કસ દેખાવ અને જગ્યા નક્કી હોય છે જે સામાન્ય રીતે વર્તમાન વર્કશીટમાં હોય છે. ઘણી વાર તમને પૂર્વનિર્ધારિત જગ્યા પસંદ ન પડે. અમુક સમયે તમને તેનું કદ ન ગમે. તમે આખો આલેખ ફક્ત માઉસનું ડાબું બટન દબાવીને ડ્રેગ કરવાથી (માઉસ વડે તેનાં હેન્ડલ દ્વારા) તમારી ઇચ્છિત જગ્યાએ ખસેડી શકો છો. તમે તેની નકલ કરી શકો, તમે તે સ્થાન પરથી દૂર કરી શકો (cut) અને અન્ય સ્થાન પર મૂકી શકો છો (paste). એ જ પ્રમાણે, આલેખ પસંદ કરી માઉસ વડે તેના હેન્ડલ ડ્રેગ કરી તેનું કદ પણ બદલી શકો છો. જુઓ આકૃતિ 8.8.

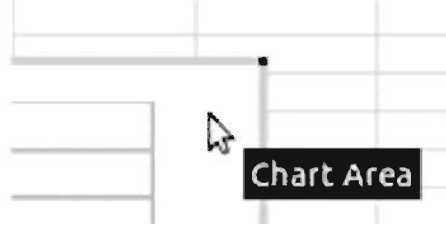


આકૃતિ 8.8 : આલેખનું કદ બદલવું

અમુક સમયે આલેખ બની ગયા પછી તમારે ડેટાની અમુક કિંમતોમાં ફેરફાર કરવાની જરૂર પડે તો ચિંતા કર્યા વગર કિંમતો બદલી નાખો. આથી આલેખ આપોઆપ બદલાઈ જશે. માઉસ વડે કોઈ સ્રોડશીટમાં આલેખ ઉમેરવામાં આવે અથવા પસંદ કરવામાં આવે તો આલેખને લગતા બટન અને આદેશ માટે કેલ્સી પેકેજનું ટૂલબાર બદલાઈ જાય છે. જ્યારે સેલમાં ડેટામાં ફેરફાર કરવામાં આવે છે તો આલેખમાં પણ ફેરફાર થાય છે.

આલેખને ફોર્મેટ કરવો જ શક્ય નથી પણ આલેખમાં સુધારા પણ કરી શકાય છે. નીચેના વિભાગમાં આલેખને ફોર્મેટ કરવા તેમજ સુધારા કરવાની રીતો વર્ણવી છે :

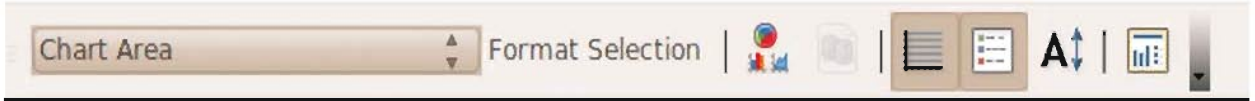
આલેખનાં કદ અને જગ્યા ઉપરાંત સૌપ્રથમ તમે કદાચ ધ્યાન ઉપર લીધો હશે તે છે તેનો પ્રકાર (type). ધારો કે column પ્રકારના આલેખમાંથી હવે તમે બદલીને line પ્રકારનો ઇચ્છો છો. આ માટે તમારે સૌપ્રથમ એ યાદ રાખવું પડે કે આલેખમાં સુધારા કરવા માટે તેને Edit modeમાં લઈ જવો પડે. આ માટે **આલેખ વિસ્તાર (Chart area)**માં કોઈ પણ જગ્યાએ ડબલ (લેફ્ટ) ક્લિક કરો. આકૃતિ 8.9માં દર્શાવ્યા પ્રમાણે આલેખની આસપાસ રાખોડી (grey) રંગની કિનારી જોવા મળશે. તમે તેનાં જુદા જુદા પ્રાયલો (parameters) બદલવા માટે આલેખ વિસ્તાર (ચાર્ટ એરિયા) ઉપર રાઈટ ક્લિક કરો.



આકૃતિ 8.9 : આલેખ વિસ્તાર (ચાર્ટ એરિયા)

આલેખનો પ્રકાર બદલવો (Modifying Chart Type)

આલેખનો પ્રકાર બદલવા માટે આ પ્રમાણે આદેશ આપો : **Format → Chart Type** આના વિકલ્પ રૂપે, આકૃતિ 8.10માં દર્શાવ્યા પ્રમાણે તમે ફોર્મેટિંગ ટૂલબાર ઉપર રહેલા **Chart type** આઈકોનને દબાવો.



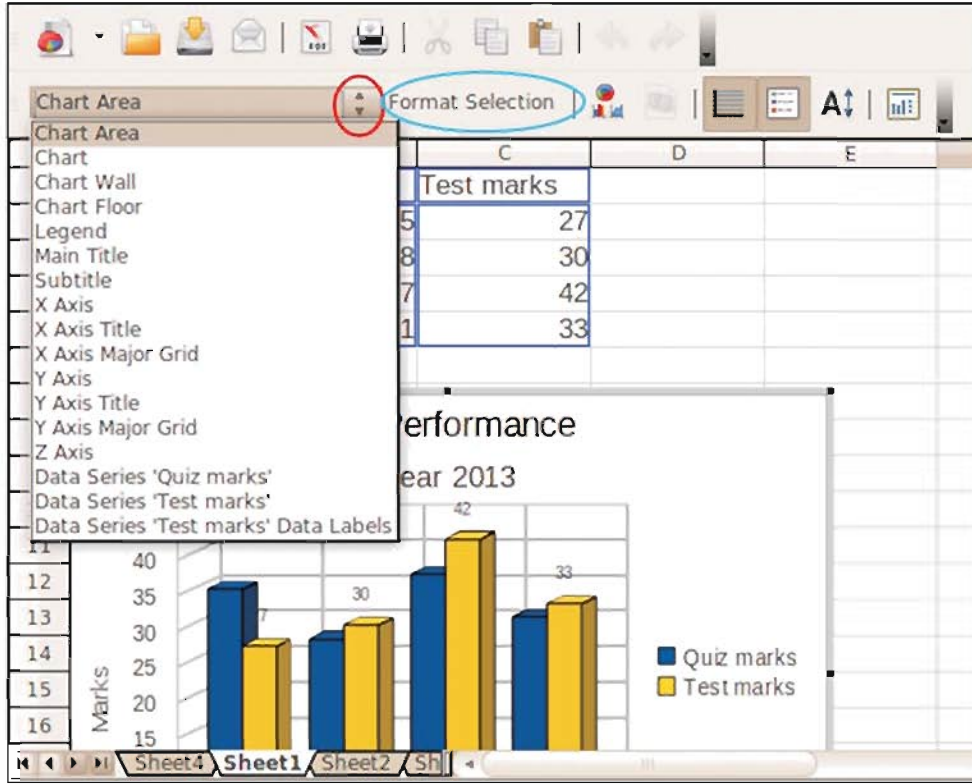
આકૃતિ 8.10 : ચાર્ટ ફોર્મેટિંગ ટૂલબાર

જ્યારે તમે આલેખ પસંદ કરો છો તે સમયે ચાર્ટ ફોર્મેટિંગ ટૂલબાર દેખાય છે. ચાર્ટ ફોર્મેટિંગ ટૂલબાર આલેખના જુદા જુદા ભાગ જેવાં કે આલેખ વિસ્તાર, આલેખની ડેટા શ્રેણી, આલેખનાં અક્ષ, આલેખનાં શીર્ષકો વગેરેને ફોર્મેટ કરવા (વધારે યોગ્ય અને આકર્ષક બનાવવા) માટે વપરાય છે. આકૃતિ 8.10માં વાદળી રંગની લીટીથી નિશાની કરેલો ત્રીજો આઈકોન આલેખના વિવિધ પ્રકાર માટેનો છે. જ્યારે તમે આ આઈકોન પર ક્લિક કરો છો ત્યારે એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર પ્રદર્શિત થાય છે. આ ડાયલોગ બોક્સના વિકલ્પો આપણને આલેખના પ્રકાર અને 3D આલેખ જેવો આલેખનો દેખાવ તેમજ બોક્સ (Box) કે પિરામિડ (Pyramid) જેવા આલેખના આકાર બદલવાની પરવાનગી આપે છે.

તમે સીધા જ આલેખ ઉપર રાઈટ ક્લિક કરીને તેનો પ્રકાર (type) બદલી શકો છો. જ્યારે તમે રાઈટ ક્લિક કરો તે સમયે સ્ક્રીન ઉપર એક નાનું ઊભું મેનૂ પ્રદર્શિત થાય છે જેમાં તે સમયે શક્ય કાર્યોની યાદી હોય છે.

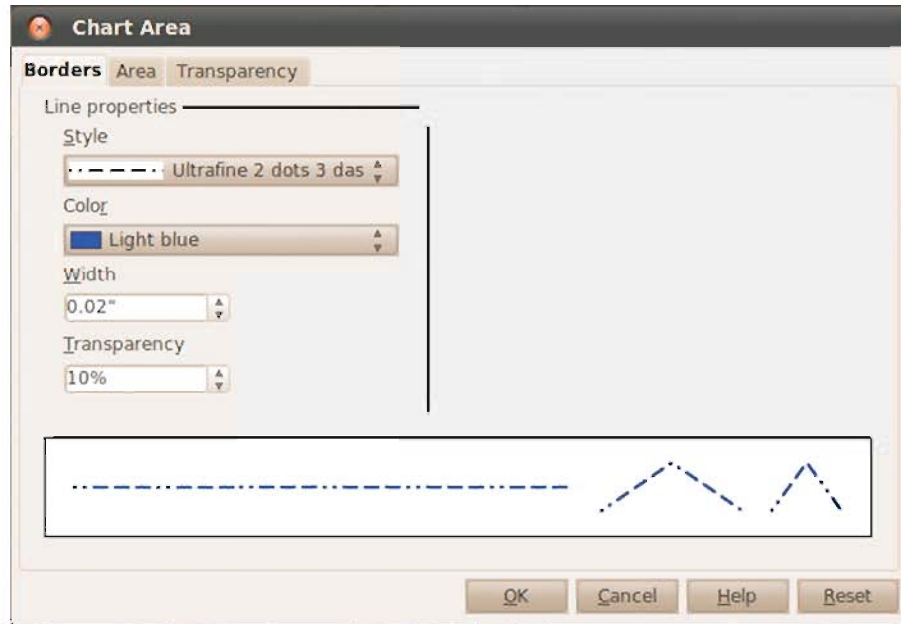
આલેખ વિસ્તારને ફોર્મેટ કરવો (Formatting the chart area)

ઉપર જણાવ્યા મુજબ આલેખ વિસ્તાર (ચાર્ટ એરિયા) એ આલેખનો એક ભાગ છે. જ્યારે તમે આલેખ ઉપર ક્લિક કરો ત્યારે આકૃતિ 8.11માં દર્શાવ્યા પ્રમાણે આલેખનું ટૂલબાર કાર્યરત થાય છે. તમે જે જે ભાગ ફોર્મેટ કરી શકો તેની યાદી ટૂલબાર ઉપરનો સૌપ્રથમ આઈકોન રજૂ કરે છે. તમે આ યાદીમાંથી જે કોઈ ભાગ ફોર્મેટ કરવા ઇચ્છતા હોય તે ચોક્કસ ભાગ પસંદ કરી શકો છો. આ માટે આકૃતિ 8.11માં દર્શાવ્યા પ્રમાણે આ બોક્સ સાથેના ઉપર અને નીચે દર્શાવતા તીર સાથેના નાના ત્રિકોણોનો ઉપયોગ કરો. આકૃતિ 8.11માં લાલ લીટીથી ચિહ્નિત (encircled) ભાગ જુઓ. એક વખત તમે યોગ્ય ભાગની પસંદગી કરી લો તો પછી **Format Selection** વિકલ્પ પસંદ કરો. આકૃતિ 8.11માં વાદળી રંગનું વર્તુળ જુઓ. આ જ આકૃતિ (8.11)માં આપણે **Chart Area** પસંદ કરેલો છે. જો કે આલેખના અન્ય ભાગોની યાદી પણ તમે જોઈ શકો છો. **Chart area** પસંદ કર્યા પછીનું પગલું **Format Selection** બટન પસંદ કરવાનું છે.



આકૃતિ 8.11 : આલેખ વિસ્તારને ફોર્મેટ કરવો

આ આપણને યોગ્ય ડાયલોગ બોક્સ વડે પસંદ કરેલાં આલેખનાં ભાગને ફોર્મેટ કરવા દે છે. આકૃતિ 8.12માં દર્શાવેલા ડાયલોગ બોક્સ આપણને આલેખના વિસ્તારની હદ (border), આલેખનો વિસ્તાર અને ભરેલા રંગની પારદર્શકતાને ફોર્મેટ કરવાની મંજૂરી આપે છે. સામાન્ય રીતે આલેખની હદ (સીમા) પૂર્વ નિર્ધારિત રીતે દૃશ્યમાન નથી પણ જો કે તમે તેને દૃશ્યમાન બનાવી શકો છો. આ ઉપરાંત સીમાની ઢબ જેમ કે સળંગ લીટી, ટપકાંવાળી લીટી, જાડી રેખા વગેરે અને સીમાનો રંગ પણ પસંદ કરી શકો છો. આલેખ વિસ્તારને વિવિધ રંગોથી, ગ્રેડિયન્ટથી (પ્રણવતા-gradient) અથવા વિવિધ પારદર્શકતાના ધોરણથી બિટ મેપ ઇમેજ (Bitmap image) વડે ભરી શકો છો. જો તમે આલેખ વિસ્તારને એક્સરખા રંગથી (પારદર્શકતા સાથે અથવા વગર) ભરવા ઇચ્છતા ન હોય તો કદાચ તમે રેખિક (linear), ત્રિજ્યાગતિ (radial) અથવા અક્ષીય (axial) દિશાઓમાં રંગ ભરવા વિચારી શકો છો.

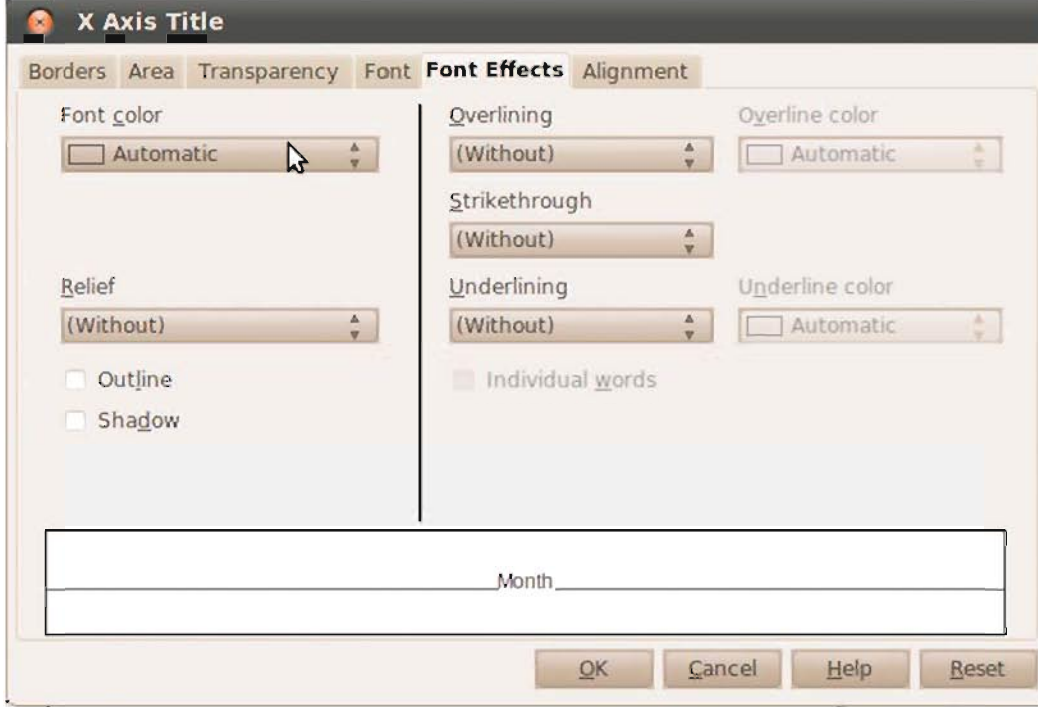


આકૃતિ 8.12 : આલેખ વિસ્તારને બદલવો

અક્ષનાં શીર્ષકોને ફોર્મેટ કરવા (Formatting titles to the axis)

આલેખને સારી રીતે સમજવા માટે X-અક્ષ અને Y-અક્ષને શીર્ષકો આપવા માટે નીચે જણાવેલી ક્રિયાઓ કરો :

- આલેખ પસંદ કરો.
- આકૃતિ 8.11માં દર્શાવ્યા પ્રમાણે ચાર્ટ ફોર્મેટિંગ ટૂલબાર તમે જોઈ શકશો. ટૂલબાર ઉપરથી X-અક્ષનું શીર્ષક ફોર્મેટ કરવા માટેનો આઈકોન પસંદ કરો.



આકૃતિ 8.13 : X-અક્ષનાં શીર્ષકો ફોર્મેટ કરવા

- હવે **Format Selection** બટન દબાવો.
- આથી તમે સ્ક્રીન ઉપર **X-Axis Titles** નામનું ડાયલોગ બોક્સ જોશો. આ ડાયલોગ બોક્સ આકૃતિ 8.13માં દર્શાવેલું છે.
- ડાયલોગ બોક્સમાં જરૂરી માહિતી ભરો.

આ પ્રકારની ફોર્મેટિંગની સગવડને કારણે તમે X-અક્ષનાં શીર્ષકોનાં અનેક ભાગોની માહિતી આપી શકો છો જેમ કે અક્ષનાં શીર્ષકના વિસ્તારની સીમારેખા, શીર્ષકના ફોન્ટ, ફોન્ટની અસરો અને તેનું એલાઈમેન્ટ (alignment ગોઠવણ). ડાયલોગ બોક્સ સીમારેખાને સેટ કરવાના વિકલ્પો, શીર્ષક વિસ્તારને ભરવા અને ભરેલા વિસ્તારની પારદર્શકતાની કક્ષા નક્કી કરવાની સગવડ પૂરી પાડે છે. અગાઉ જણાવ્યા પ્રમાણે તમે X-અક્ષની કિંમતોના ફોન્ટ ફોર્મેટ કરી શકો અને ફોન્ટની અસરો બદલી પણ શકો છો. **Font** ટેબ તમને યોગ્ય ફોન્ટ પસંદ કરવા દે છે. જો તમારા કમ્પ્યુટરમાં યોગ્ય ફોન્ટ હોય તો તમે ગુજરાતી અને હિન્દી જેવી પ્રાદેશિક ભાષામાં પણ લેબલ પ્રદર્શિત કરી શકો છો. તમે ફોન્ટના રંગ, વિવિધ રંગની રેખાઓ વડે શીર્ષકોના ફોન્ટની રેખા ઢાંકવી, અક્ષર લખીને ઉપર લીટી દોરવી, વિવિધ રંગની રેખાઓ વડે શીર્ષકોના ફોન્ટ નીચે લીટી દોરવી વગેરે જેવી ફોન્ટની અસરો સામેલ પણ કરી શકો અને દૂર પણ કરી શકો છો. તમે શીર્ષકના ફોન્ટને આકર્ષક અને સુશોભિત દેખાવ આપવા માટે Shadow અને Outline અસરો પણ આપી શકો છો. આ ઉપરાંત આકૃતિ 8.13માં દર્શાવ્યા પ્રમાણે તમે ડાયલોગ બોક્સના અંતિમ ટેબ કે જેને **Alignment** નામ આપેલ છે તેને પસંદ કરીને તમે શાબ્દિક લખાણનું વહેણ (flow) તેમજ શાબ્દિક લખાણની ગોઠવણ (alignment) પણ કરી શકો છો. આ જ પ્રમાણે તમે Y-અક્ષનાં શીર્ષકોને પણ ફોર્મેટ કરી શકો છો.

X-અક્ષને ફોર્મેટ કરવી (Formatting X-Axis)

આલેખનાં ફોર્મેટિંગ ટૂલબાર વડે તમે X-અક્ષને પણ ફોર્મેટ કરી શકો છો. પ્રથમ યોગ્ય ભાગ પસંદ કરો, અહીં X-Axis પસંદ કરો અને પછી **Format Selection** બટન દબાવો. આથી સ્ક્રીન ઉપર તમે એક ડાયલોગ બોક્સ જોશો. આ ડાયલોગ

બોક્સનો ઉપયોગ કરીને તમે X-અક્ષના માપક્રમ (Scale) માટેના વિકલ્પો સુયોજિત કરી શકો છો કે જે સામાન્ય રીતે આપોઆપ થાય છે. તમે અક્ષનું સ્થાન એટલે કે અન્ય અક્ષ સાથે શરૂઆત, અંત અથવા અન્ય જગ્યાએ ક્યાં છેદાય તે સેટ કરી શકો છો. અક્ષનાં સ્થાન સાથે તમે અક્ષ ઉપર ક્યાં લેબલ ઈચ્છો છો તે પણ જણાવી શકો છો. તમે અક્ષની નજીક, અક્ષની બહાર, અક્ષની શરૂઆતમાં અથવા અક્ષનાં અંતમાં લેબલ રાખી શકો છો.

અનેક સમયે લેબલ ઘણા વર્ણનાત્મક હોય છે જેના કારણે આલેખનાં કદનું સંચાલન મુશ્કેલ બને છે અને એ સમયે તમે ફોન્ટનું કદ નાનું કરવા ઈચ્છો. જો તમારે લેબલની જરૂર જ બિલકુલ ન હોય તો **Show labels** વિકલ્પ ઉપલબ્ધ છે. તમે ખરાની નિશાની કરી તે વિકલ્પને નાપસંદ કરી શકો, જે તે અક્ષનાં લેબલ કાઢી નાખશે. આ જ પ્રમાણે તમે Y-અક્ષને પણ ફોર્મેટ કરી શકો.

ચાર્ટ વોલ ફોર્મેટ કરવી (Formatting the chart wall)

ચાર્ટ વોલને ટૂલબાર ઉપરના વિકલ્પો પસંદ કરી અને **Format Selection** બટન પસંદ કરીને ફોર્મેટ કરી શકાય છે. આ કાર્યથી એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર દેખાશે જે આકૃતિ 8.11માં દર્શાવેલ છે તેના જેવું છે. આ ડાયલોગ બોક્સ તમને સીમારેખા(બોર્ડર)ના ગુણધર્મો જેવા કે સીમારેખા દર્શ્યમાન કે અદર્શ્ય રાખવી, સીમારેખાનો પ્રકાર, સીમારેખાનો રંગ, સીમારેખાની જાડાઈ વગેરે બદલવાની સગવડ પૂરી પાડે છે. સીમારેખાના ગુણધર્મો ઉપરાંત આ ડાયલોગ બોક્સ તમને વિસ્તારનો રંગ અને પારદર્શકતાનું પ્રમાણ બદલવાની તક પૂરી પાડે છે.

ચાર્ટ ફ્લોર ફોર્મેટ કરવી (Formatting the chart floor)

ચાર્ટ ફ્લોર પણ ટૂલબાર ઉપરના વિકલ્પો પસંદ કરીને **Format Selection** બટન પસંદ કરીને બદલી શકાય છે. ચાર્ટ ફ્લોર ફોર્મેટિંગ વિકલ્પ તમે આલેખ પસંદ કરી, ચાર્ટ ફ્લોર ઉપર રાઈટ ક્લિક કરી અને પોપ-અપ મેનૂમાંથી **Chart floor** વિકલ્પ પસંદ કરીને કરી શકો છો. આ કાર્ય કરવાથી ડાયલોગ બોક્સ સ્ક્રીન ઉપર દેખાશે જે તમને આલેખની સીમારેખા, વિસ્તાર અને પારદર્શકતાનું પ્રમાણ જેવા ગુણધર્મો સુધારવાની સગવડ આપે છે.

આલેખનાં લિજેન્ડ ફોર્મેટ કરવાં (Formatting chart legends)

જો તમે લિજેન્ડ દાખલ કરવાનું સંપૂર્ણપણે ભૂલી ગયા હોય અથવા આલેખ બનાવતા સમયે છોડી દીધા હોય તો તમે આલેખ ઉપર રાઈટ ક્લિક કરી અને **Insert Legend** વિકલ્પ પસંદ કરીને ઉમેરી શકો છો. એક સમયે લિજેન્ડ તમને જોવામાં આવે તે પછી તમે તેને સુધારવાનું વિચારી શકો. આકૃતિ 8.11માં દર્શાવ્યા પ્રમાણે આલેખનાં ટૂલબાર વડે લિજેન્ડને તમે ફોર્મેટ કરી શકો છો. જ્યારે તમે આલેખનાં ફોર્મેટિંગ ટૂલબારમાંથી ફોર્મેટિંગ ચાર્ટ લિજેન્ડ પસંદ કરી **Format Selection** બટન પર ક્લિક કરો છો ત્યારે હંમેશા મુજબ એક ડાયલોગ બોક્સ સ્ક્રીન ઉપર દેખાશે. જો તમારા ધ્યાનમાં આવ્યું હોય તો આલેખનાં લિજેન્ડ પણ અદર્શ્ય સીમારેખા સાથે લંબચોરસમાં ગોઠવાયેલાં હોય છે. આ ડાયલોગ બોક્સનો ઉપયોગ કરીને તમે લિજેન્ડ વિસ્તારની સીમારેખા ગોઠવી શકો, વિસ્તારને ફોર્મેટ કરી શકો અને પારદર્શકતાનું પ્રમાણ બદલી શકો વગેરે. આ વિકલ્પો અગાઉ સમજાવ્યા પ્રમાણે જ કામ કરે છે. આ વિકલ્પો ઉપરાંત વધારાના અન્ય વિકલ્પો જેવા કે ફોન્ટ, ફોન્ટની અસરો અને આલેખના સ્થાન વિસ્તારમાં રહેલાં લિજેન્ડની જગ્યા પણ ડાયલોગ બોક્સમાં પૂરા પાડવામાં આવ્યા છે. આ વિકલ્પો જરૂરી છે કારણ કે આલેખનાં લિજેન્ડમાં કેટલીક શાબ્દિક માહિતી સાથે થોડાં ગ્રાફિક્સ (ચિત્રો) પણ હોય છે. આ ઉપરાંત લિજેન્ડની શાબ્દિક માહિતી આલેખમાં કઈ જગ્યાએ મૂકવાની છે તે પણ જણાવવું જરૂરી છે. જો કંઈ ન જણાવવામાં આવે તો આલેખની જમણી બાજુની જગ્યાએ રાખવામાં આવે છે પણ જો કે તમે લિજેન્ડને આલેખની ઉપર, આલેખની નીચે અથવા આલેખની ડાબી બાજુએ પણ રાખી શકો છો. તમે કોઈ પણ અનુકૂળ ફોન્ટને તેના કદ અને દેખાવ જેમ કે ઘાટો, ત્રાંસો કે લખાણ નીચે લીટી દોરેલો પસંદ કરી શકો છો.

ડેટાની શ્રેણી ફોર્મેટ કરવી (Formatting data series)

ડેટાની શ્રેણી બદલવા માટે તમે આલેખ પસંદ કરી તેના ઉપર રાઈટ ક્લિક કરો. એક ઊભું મેનૂ પ્રદર્શિત થશે. મેનૂમાંથી **Format Data Series** વિકલ્પ પસંદ કરો અને તે પછી તેને ફોર્મેટ કરો. આ જ કાર્ય બીજી રીતે કરવા માટે આલેખનાં ફોર્મેટિંગ ટૂલબાર (આકૃતિ 8.11માં દર્શાવ્યા પ્રમાણે) પરથી **Data Series** આઈટમ પસંદ કરો અને **Format Selection** બટન દબાવો. આ તમને ડાયલોગ બોક્સ દ્વારા ડેટા શ્રેણીનાં પ્રાયલો (parameters) બદલવાની તક આપશે.

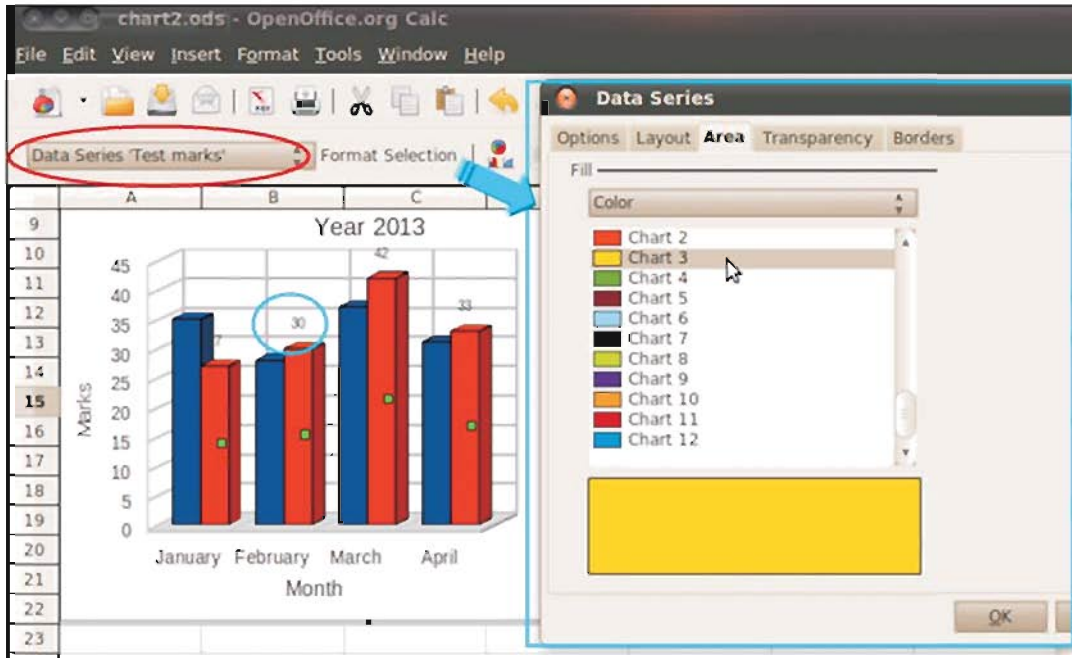
ડેટાની શ્રેણીને ફોર્મેટ કરવાની અન્ય રીત નીચે વર્ણવી છે :

- આલેખ તૈયાર થયા પછી ડેટા શ્રેણી નિરૂપણ કરતી કોઈ પણ કોલમ (ઊભી હરોળ) ઉપર ક્લિક કરો. જો તમે એક લાલ રંગની કોલમ ઉપર ક્લિક કર્યું હશે તો તમે નિરીક્ષણ કરી શકશો કે બધી લાલ રંગની કોલમ કાર્યરત થયેલી છે. આ ડેટા શ્રેણી રજૂ કરે છે.
- આકૃતિ 8.14માં લાલ રંગના કોલમવાળી વર્ગપરીક્ષાના ગુણની ડેટા શ્રેણી પસંદ કરેલી પદ્ધતિથી દર્શાવી છે.
- આલેખનો ફોર્મેટિંગ ટૂલબાર શીટમાં ઉપરની જગ્યાએ દેખાય છે. લાલ વર્તુળમાં આપેલો ભાગ જુઓ. તે **Data series 'Test Marks'** દેખાડે છે.
- **Format Selection** બટન દબાવો. તમે આલેખ સાથે બીજું એક લંબચોરસ ડાયલોગ બોક્સ જોઈ શકશો. બહારથી વાદળી રંગની રેખા સાથેનો લંબચોરસ વિસ્તાર જુઓ.
- તમે layout, area filling, transparency level વગેરેમાંથી યોગ્ય ટેબ (tab) પસંદ કરો.
- આકૃતિ 8.14માં દર્શાવ્યા પ્રમાણે તમે જોઈ શકશો કે આપણે fill colour લાલથી પીળામાં ફેરવીએ છીએ તે પ્રમાણે આલેખના વિસ્તારનો દેખાવ બદલાય છે.

આ જ પ્રમાણે તમે બાકીની ડેટા શ્રેણીને પણ ફોર્મેટ કરી શકો છો.

ડેટા લેબલ ઉમેરવા (Inserting data labels)

બીજું એક આ જ પ્રકારનું કાર્ય આલેખના દરેક ડેટા કોલમ સાથે ડેટા લેબલ દાખલ કરવાનું છે. ધારો કે તમે આલેખના કોઈ ભાગ જેમ કે કોલમ સાથે ખરેખરી કિંમત ઇચ્છો છો તો તમારે ફક્ત ડેટા શ્રેણીને કાર્યરત બનાવી રાઈટ ક્લિક જ કરવાની છે. તે પછી તમે **Insert Data Labels** પસંદ કરો. આકૃતિ 8.14માં તમે જોઈ શકશો કે ડેટા લેબલ અત્યાર સુધીમાં ઉમેરાઈ ગયા છે. આકૃતિ 8.14માં વર્તુળથી નિશાની કરેલી વાદળી રેખા સાથેની સંખ્યા જુઓ. આ જ પ્રમાણે અન્ય ડેટા શ્રેણીને તમે ફોર્મેટ કરી શકો છો.



આકૃતિ 8.14 : ડેટા શ્રેણીને ફોર્મેટ કરવી

વાસ્તવિક ઉદાહરણ (Working example)

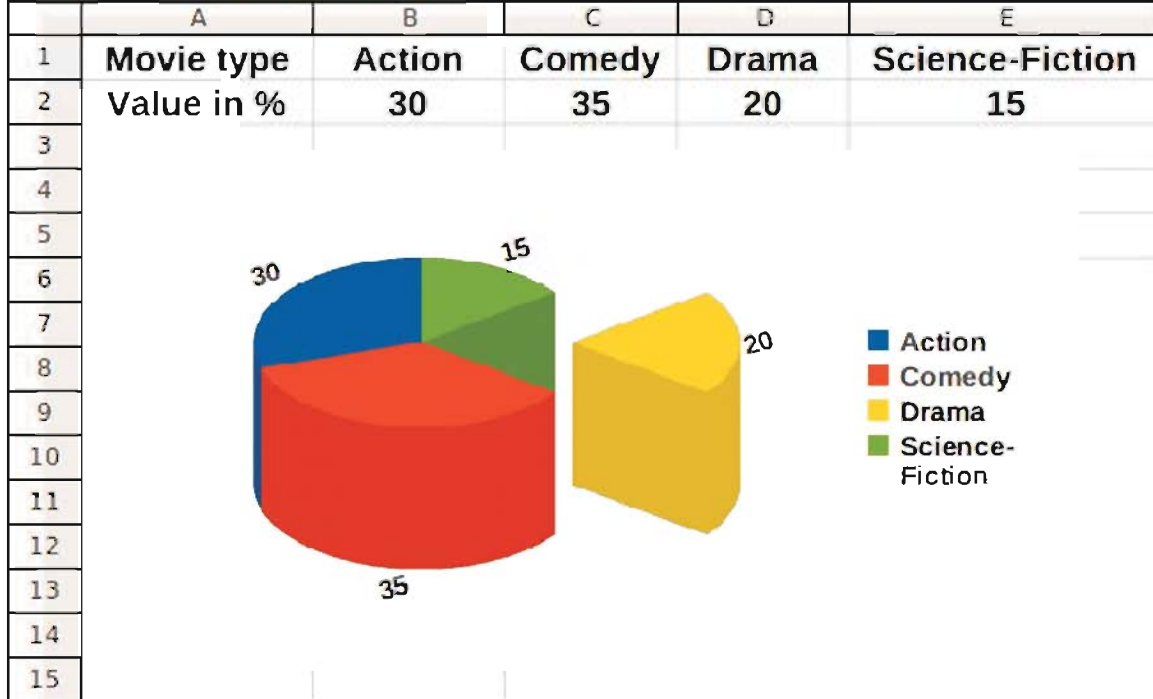
ચાલો, આપણે બીજા પ્રકારનો એક આલેખ તૈયાર કરીએ. નીચે આપેલા ડેટા લો. ધારો કે તે ડેટા ફિલ્મ(movies)ને લગતા છે. તમને બધાને ફિલ્મનો આનંદ માણવો ગમતો હશે. જો કે દરેક વ્યક્તિની અલગ પસંદ અને રુચિ હોય છે. કેટલાકને action મૂવી પસંદ હશે જ્યારે કેટલાકને Comedy પણ Science Fiction ફિલ્મનો એક વિશિષ્ટ પ્રેક્ષકવર્ગ હોય છે. આ સ્થિતિને વર્ણવતો ડેટા કોષ્ટક 8.2માં આપેલો છે.

Movie type	Action	Comedy	Drama	Science-Fiction
Value in %	30	35	20	15

કોષ્ટક 8.2 : અલગ અલગ ફિલ્મની પસંદ

કોષ્ટક 8.2માં આપેલી માહિતીને આધારે આપણે એક આલેખ બનાવીએ. નીચે જણાવેલ કાર્ય કરો :

- કેલ્સી વર્કશીટમાં કોષ્ટક 8.2માં આપેલા ડેટા દાખલ કરો અને યોગ્ય નામ આપી તેનો સંગ્રહ (save) કરો.
- ખાતરી કરો કે ડેટાનો વિસ્તાર A1થી E2 છે.
- **Insert → Chart** આદેશ આપો.
- Chart typeમાંથી **Pie** પ્રકારનો આલેખ પસંદ કરો.
- Lookમાં **3D** અને **Simple** પસંદ કરો.
- **Finish** બટન દબાવો.
- આથી આકૃતિ 8.15માં દર્શાવ્યા મુજબનો આલેખ તમે જોઈ શકશો.
- જો કે તમારા આલેખમાં ડેટા લેબલ દેખાતા નથી તેમજ Drama શ્રેણી બહાર પણ કાઢેલી નથી. ડેટા લેબલ ઉમેરવા તેમજ Dramaની શ્રેણી (પીળા રંગથી નિર્દેશિત) બહાર કાઢવા માટે નીચે આપેલાં પગલાં પ્રમાણે કાર્ય કરો :
- આલેખ પસંદ કરો. આથી તમે રાખોડી રંગની સીમારેખા આલેખની આસપાસ જોઈ શકશો.
- આલેખ ઉપર રાઈટ ક્લિક કરો, જેથી એક ઊભું મેનૂ પ્રદર્શિત થશે.
- હવે **Insert Data Labels** પસંદ કરો.



આકૃતિ 8.15 : પાઈ ચાર્ટ (Pie chart)

- તમે જોઈ શકશો કે આલેખ સાથે કિંમતો દર્શાવાયેલી છે.
- પાઈ ચાર્ટમાંથી એક ભાગ બહાર કાઢવા માટે માઉસ ક્લિક કરો. ત્યાર બાદ માઉસ વડે તે ભાગને ઈચ્છિત જગ્યા સુધી ખસેડો (ડ્રાગ કરો).

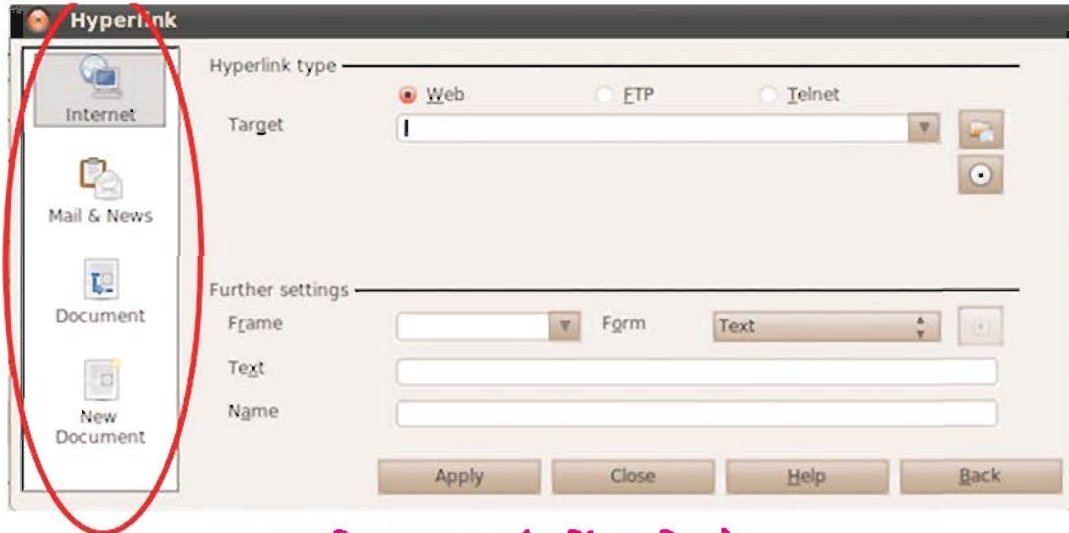
- જો તમે વધારે ફોર્મેટિંગ કરવા ઇચ્છતા હો તો આલેખ પસંદ કરો. આલેખનું ફોર્મેટિંગ ટૂલબાર તમને દેખાશે. તમારે જે ભાગનું ફોર્મેટિંગ કરવું હોય તે પસંદ કરી **Format Selection** આપો. આ સગવડથી તમે આલેખનો પ્રકાર (ટાઇપ), આલેખ વિસ્તાર (એરિયા), ડેટા શ્રેણી, ફોન્ટ, ફોન્ટ અસરો, અક્ષનો દેખાવ, લિજેન્ડ, ગ્રીડ, ચાર્ટ ફ્લોર અને વોલ વગેરેમાં ફેરફાર કરી શકો છો. અહીં આપણે ફોન્ટને ઘાટા (બોલ્ડ) કરી શકીએ તેમજ ફોન્ટનું કદ વધારી પણ શકીએ છીએ. આપણે ડેટા લેબલ 15 અંશ વર્તુળાકાર ફેરવેલાં પણ છે.
- આ જ પ્રમાણે તમે આલેખનાં લિજેન્ડ ફોર્મેટ કરી શકો છો અને લિજેન્ડના ફોન્ટનું કદ વધારી શકો છો.

આલેખ સાથે હાઇપરલિંક જોડવી (Adding hyperlink to the chart)

હાઇપરલિંક (Hyperlink) એ ડેટાનો સંદર્ભ છે જે પસંદ કર્યા પછી તેને અનુસરી શકાય છે. તમે જાણો છો તે પ્રમાણે હાઇપરલિંક કોઈ આખા દસ્તાવેજ કે દસ્તાવેજના ભાગને ચીંધે છે. એ શબ્દિક લખાણ કે જે આવા જોડાણ(link) સમાવેશ કરે છે તેને હાઇપરટેક્સ્ટ (hypertext) કહેવામાં આવે છે. એક વાર આલેખ તૈયાર થઈ જાય પછી તમે આખા આલેખને અથવા આલેખના કોઈ ભાગ સાથે હાઇપરલિંક જોડી શકો છો. જ્યારે વપરાશકર્તા હાઇપરલિંક ઉપર ક્લિક કરે છે ત્યારે તે પૂર્વનિર્ધારિત જગ્યાએ વપરાશકર્તાને લઈ જાય છે. તમે વેબ ઉપર રહેલાં કોઈ દસ્તાવેજ, તમારા લોકલ એરિયા નેટવર્ક / કમ્પ્યૂટરના દસ્તાવેજ કે નવા દસ્તાવેજ સાથે હાઇપરલિંક જોડી શકો છો.

કોઈ પણ પ્રકારની હાઇપરલિંક માટે તમારી પાસે આલેખ હોવો જોઈએ. તમારે સૌપ્રથમ આલેખ તૈયાર કરવો જોઈએ. કોઈ વેબ (ઇન્ટરનેટ) દસ્તાવેજ સાથે હાઇપરલિંક બનાવવા માટે નીચે જણાવેલ પગલાંને અનુસરો :

- સૌપ્રથમ ડેટા દાખલ કરો અને યોગ્ય પ્રકારનો આલેખ બનાવો.
- આલેખને ચકાસો અને માઉસ ક્લિક વડે પસંદ કરો.
- **Insert → Hyperlink** આદેશ આપો.
- તે તમને ગંતવ્ય સ્થાનની લિંક માટે ચાર વિકલ્પો આપશે. તમે આલેખ(અથવા પસંદ કરેલા ભાગ)ને Internet, Mail & News, Document અથવા New Document સાથે હાઇપરલિંક કરી શકો છો. આકૃતિ 8.16માં આલેખને જોડવા માટેના ચાર વિકલ્પો (લાલ અંડાકારથી ચિહ્નિત કરેલાં) દર્શાવ્યા છે.



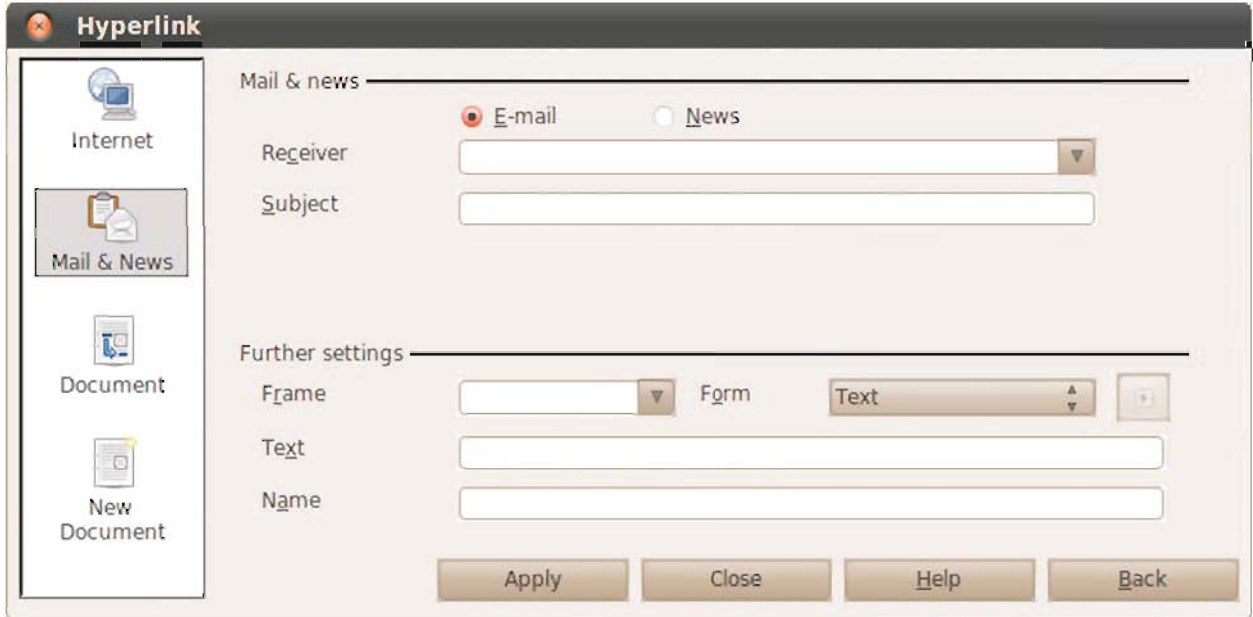
આકૃતિ 8.16 : હાઇપરલિંકના વિકલ્પો

પૂર્વનિર્ધારિત રીતે પહેલો વિકલ્પ **"Internet"** હાઇલાઇટ થયેલો હોય છે અને **Target**માં **Google.com**. જેવું અંતિમ મુકામ URL આપી શકો છો. જો તમારે આ લિંકનો પ્રયોગ કરવો હોય તો ઇન્ટરનેટ જોડાણ હોવું જરૂરી છે. જો બધું બરાબર હોય તો ક્લિક કરવાથી આલેખ તમને ગૂગલનાં મુખ્ય પાના ઉપર લઈ જશે. આના વિકલ્પરૂપે કોઈ પણ પ્રકારની હાઇપરલિંક **Ctrl + Enter** દબાવીને અનુસરી શકાય છે.

તમે બધી જરૂરી માહિતી દાખલ કરી દીધા પછી **Apply** પસંદ કરી શકો છો. ડાયલોગ બોક્સ બંધ કરવા માટે **Close** બટનનો ઉપયોગ કરો.

મેઈલ સાથે આલેખ જોડવો (Linking a chart to a mail)

જો તમારે મેઈલ એન્ડ ન્યૂઝ સાથે આલેખ જોડવો હોય તો **Mail and News** વિકલ્પ પસંદ કરો. તમે એક અલગ ગ્રાફલોગ બોક્સ સ્ક્રીન ઉપર જોશો. જ્યારે તમે **Mail & News** સાથે જોડાણ કરો છો ત્યારે આકૃતિ 8.17માં દર્શાવ્યા પ્રમાણેના વિકલ્પો જોવા મળશે.

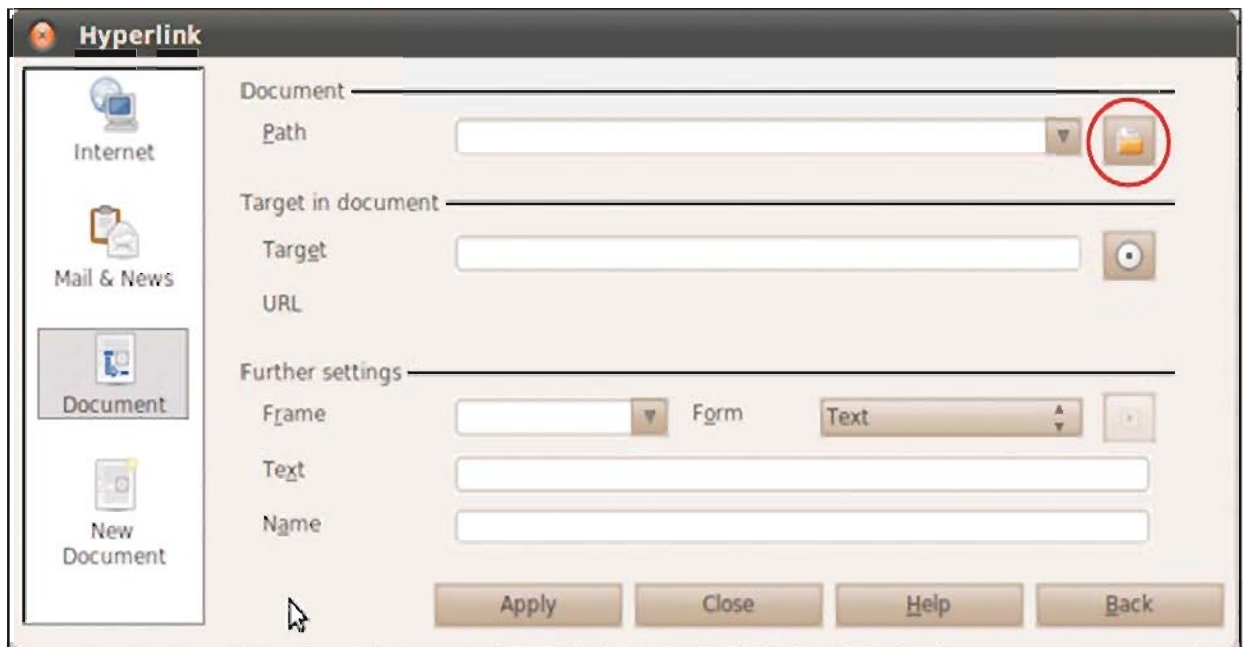


આકૃતિ 8.17 : હાઈપરલિંકમાં Mail and Newsના વિકલ્પો

તમારે મેઈલનાં receiver અને subjectની માહિતી આપવી પડશે. જ્યારે તમે **Apply** વિકલ્પ આપશો ત્યારે તે આલેખ મેળવનારનાં સરનામાં ઉપર ટપાલ માટે ક્તારમાં હશે. જો કે આ પ્રક્રિયા ઓફલાઈન રહે છે અને મેઈલ ક્લાયન્ટ (mail client) પશ્ચાદ્ભૂમિમાં રાખવાની જરૂર રહે છે. એ જ પ્રમાણે, **News**નાં વિકલ્પ માટે તમારે ન્યૂઝ સર્વર(news server)ની રચના (configure) કરવાની જરૂર પડે છે. અંતમાં **Close** બટન દબાવો.

દસ્તાવેજ સાથે આલેખ જોડવો (Linking a chart to a document)

દસ્તાવેજ સાથે આલેખ જોડવા માટે ત્રીજો વિકલ્પ Document પસંદ કરો. આથી તમે આકૃતિ 8.18માં દર્શાવ્યા પ્રમાણેનો સ્ક્રીન જોશો.



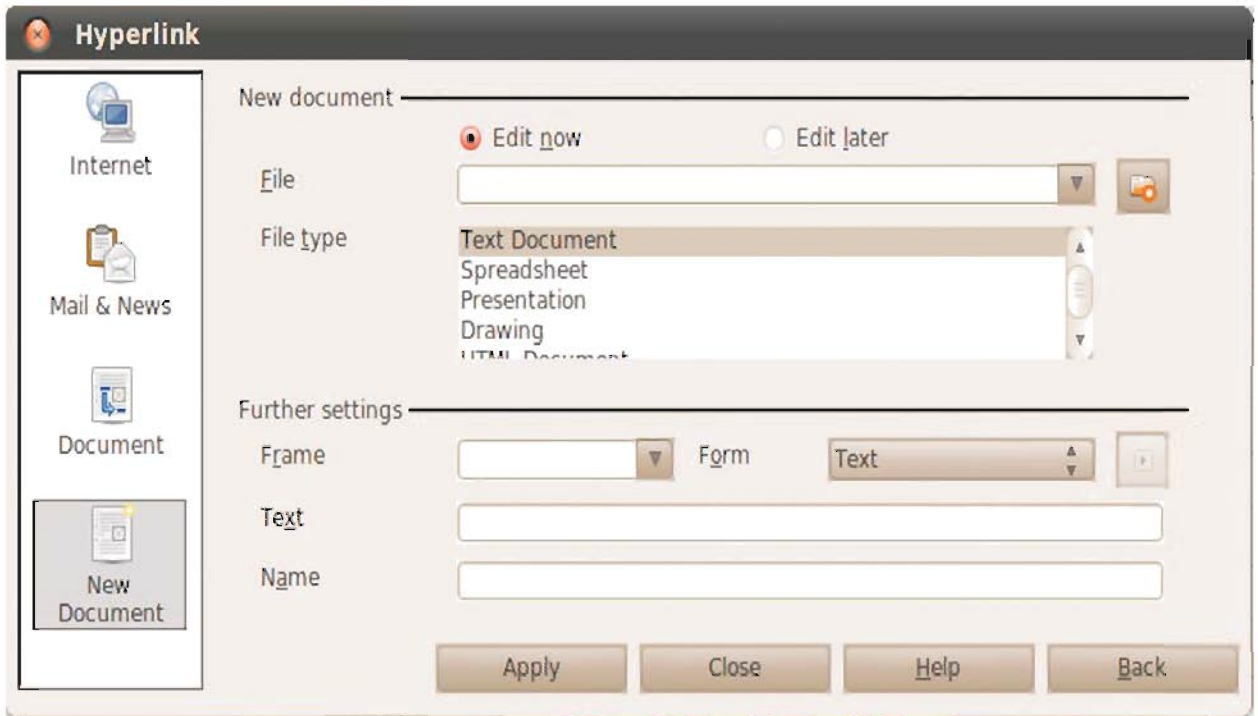
આકૃતિ 8.18 : દસ્તાવેજ સાથે આલેખને જોડવો

જ્યારે તમે નાનો આલેખ બનાવ્યો હોય ત્યારે આ વિકલ્પ ઉપયોગી છે કે જેમાં વધારે સમજણ બીજા દસ્તાવેજમાં આપેલી હોય છે. આવા સંજોગોમાં વધારે માહિતી માટે તમે આલેખને અન્ય દસ્તાવેજ સાથે જોડો. ટૂંકમાં તમે એમ વિચારો કે અમુક દસ્તાવેજ આલેખ સાથે સંબંધિત છે તો તમે આ સગવડ દ્વારા આલેખ સાથે જોડી શકો છો. દસ્તાવેજનો સંપૂર્ણ પાથ (path) અંતિમ સ્થાન સૂચવે છે. જો તમને સંપૂર્ણ પાથ યાદ ન હોય તો તમે pathની બાજુમાં રહેલા શાબ્દિક બોક્સની પાસેના Browse બટન(લાલ લીટીથી ગોળ ચિહ્નિત) નો ઉપયોગ કરીને દસ્તાવેજને જોઈ શકો છો (બ્રાઉઝ કરી શકો છો). તમે **Target** વિકલ્પનો ઉપયોગ કરીને અંતિમ મુકામના દસ્તાવેજમાં લક્ષ (target) પણ આપી શકો છો.

અંતમાં **Apply** બટન પર ક્લિક કરો. ડાયલોગ બોક્સ બંધ કરવા માટે અને કાર્યની સમાપ્તિ માટે **Close** બટન દબાવો.

નવા દસ્તાવેજ સાથે આલેખ જોડવો (Linking a chart to a new document)

નવા દસ્તાવેજ સાથે આલેખને જોડવા માટે ચોથો વિકલ્પ **New Document** પસંદ કરો. તમે **File** અને **File type** આપો. તમે આ વિકલ્પ વડે સ્પ્રેડશીટ, વર્ડ પ્રોસેસર, ડ્રોઈંગ વગેરે જેવી વિવિધ પ્રકારની ફાઈલ બનાવી શકો છો. એક વાર ફાઈલ બની જાય તે પછી તેમાં સુધારા-વધારા કરવા માટે તે ઉપલબ્ધ બની જાય છે અને પૂર્વનિર્ધારિત રીતે **Edit now** વિકલ્પ પસંદ થયેલો છે. જો તમે હમણાં કોઈ પણ પ્રકારના સુધારા કરવા ન ઇચ્છતા હોય તો **Edit later** વિકલ્પ તમે પસંદ કરી શકો છો. ઉપર જણાવેલી સ્થિતિ આકૃતિ 8.19માં દર્શાવેલી છે.



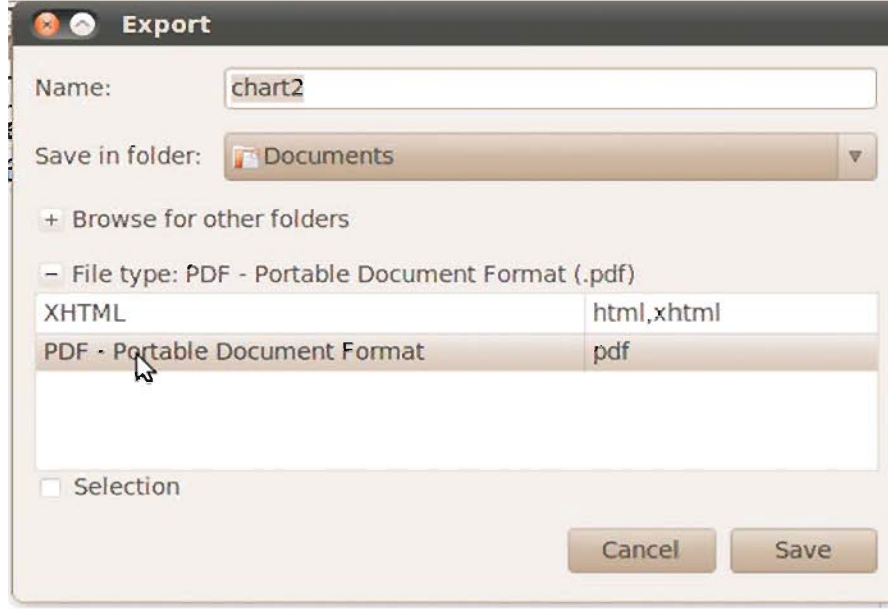
આકૃતિ 8.19 : નવા દસ્તાવેજ સાથે આલેખ જોડવો

અંતમાં, **Apply** વિકલ્પ આપો. ડાયલોગ બોક્સ બંધ કરવા માટે **Close** બટનનો ઉપયોગ કરો.

આલેખ નિકાસ કરવો અને અન્ય કાર્યો (Exporting the chart and other operations)

તમે આલેખ સાથેના દસ્તાવેજને PDF અનુલંબન સાથેની સુવાહ્ય (portable) દસ્તાવેજ ફાઈલ તરીકે નિકાસ કરી શકો છો. આ કાર્ય માટે નીચે પ્રમાણે પગલાં ભરો :

- સૌપ્રથમ **File → Export** આદેશ આપો. આથી આકૃતિ 8.20માં દર્શાવ્યા પ્રમાણેનો સ્ક્રીન તમે જોઈ શકશો.

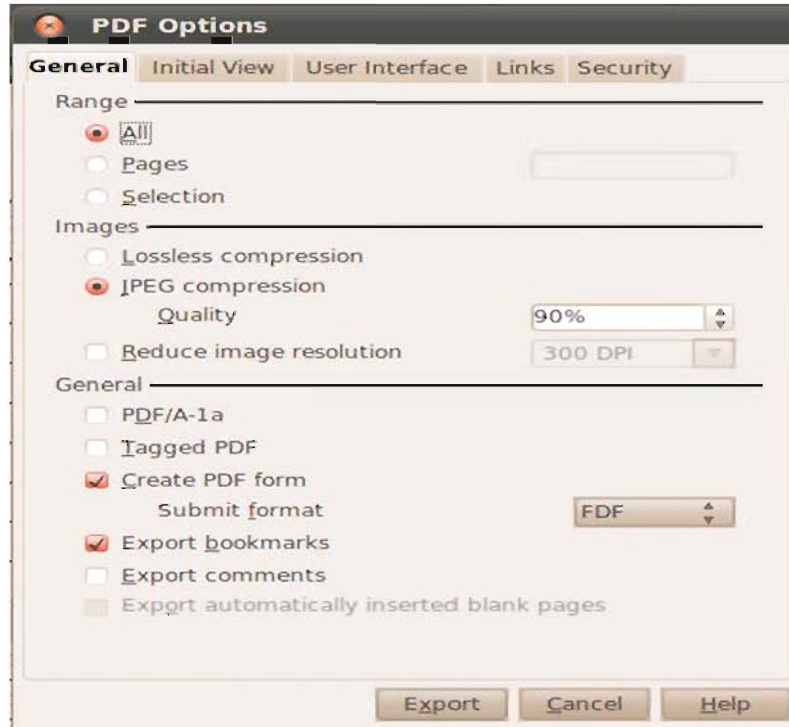


આકૃતિ 8.20 : દસ્તાવેજની નિકાસ કરવી

ફાઈલની નિકાસ XHTML (વેબ પેઈજ) અથવા PDF (પોર્ટેબલ ડોક્યુમેન્ટ ફાઈલ) જેવા બે વિકલ્પોમાં કરી શકો તે તમને જોવા મળશે.

- **PDF** પસંદ કરો.
- તમારા ઈચ્છિત ફોલ્ડરમાં ફાઈલનો સંગ્રહ કરો.

જ્યારે તમે તમારી પસંદ PDF તરીકે આપો છો ત્યારે આકૃતિ 8.21માં દર્શાવ્યા પ્રમાણે અનેક વિવિધ પ્રાયલો(પેરામીટર)ની માહિતી લઈ આખરે PDF ફાઈલમાં પરિણમતી ફાઈલ માટેનું ડાયલોગ બોક્સ તમે જોશો. આકૃતિ 8.21માં દર્શાવેલા ડાયલોગ બોક્સનો ઉપયોગ કરીને વિવિધ વિકલ્પો જેમ કે પાનાંની સંખ્યા, ઈમેજ(ગ્રાફિક્સ)નું ફોર્મેટ (જો તેમાં સમાવેલ હોય તો) વગેરેને નિયોજિત કરી શકાય છે. જો તમે અમુક પાનાંની જ નિકાસ કરવા માટે પસંદ કરેલાં હોય તો PDFમાં પરિણમતા દસ્તાવેજમાં ફક્ત તે જ પાનાંની નિકાસ થશે. પૂર્વ નિર્ધારિત રીતે તે આખો દસ્તાવેજ હોય છે.



આકૃતિ 8.21 : દસ્તાવેજની PDF સ્વરૂપમાં નિકાસ કરવી

આ ઉપરાંત દસ્તાવેજની બધી ઈમેજ jpeg સ્વરૂપમાં સંગ્રહ થાય છે. અહીં તમારી પાસે નુકસાનરહિત ઈમેજ સંકોચન અને આપેલા ગુણવત્તાના ટકા સાથે JPEG ઈમેજ સંકોચનના વિકલ્પો હોય છે. જો તમે પસંદ ન કરેલો હોય તો **Create PDF** વિકલ્પ પસંદ કરો. સામાન્ય રીતે આ બધા વિકલ્પો પૂર્વનિર્ધારિત હોય છે અને કમ્પ્યુટર (સિસ્ટમ) દ્વારા પોતાની જાતે જ પસંદ કરી લેવામાં આવે છે. તમે Generalના પૂર્વ નિર્ધારિત વિકલ્પોમાં કોઈ ફેરફાર ન પણ કરો. તમે સીધા જ **Export** બટન દબાવો. આ કરવાથી આલેખ સાથેનો દસ્તાવેજ PDF સ્વરૂપમાં નિકાસ થશે.

આલેખની નકલ કરવી (Copying the chart)

આલેખની નકલ અન્ય પ્રોગ્રામ જેવા કે Writer કે Impressમાં કરવા માટે તમારે તે આલેખ પસંદ કરી કોપી (copy) કરો અને જ્યાં જરૂર હોય તે જગ્યાએ પેસ્ટ (paste) કરો. આલેખની નકલ કરવા માટે નીચે જણાવેલાં સરળ પગલાં પ્રમાણે કાર્ય કરો :

- માઉસ ક્લિક કરી વડે આલેખ પસંદ કરો.
- રાઈટ ક્લિક કરીને **Copy** અથવા **Cut** વિકલ્પ આપો.
- તમે **Edit** મેનૂમાં જઈને જરૂરી વિકલ્પ પણ પસંદ કરી શકો છો.
- હવે જેમાં નકલ કરવાની છે તે દસ્તાવેજમાં જાઓ. હવે **Paste** આદેશ આપો.
- જેમાં નકલ કરી તે ફાઈલનો સંગ્રહ (save) કરો.

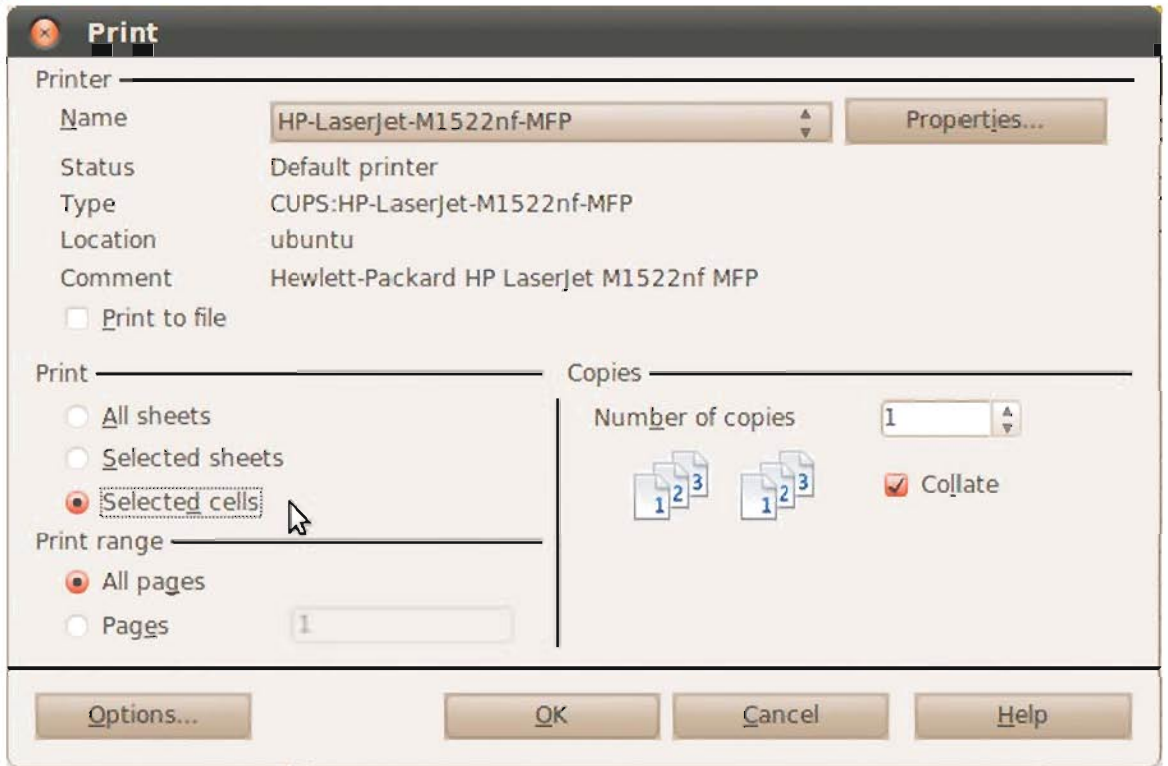
આલેખ દૂર કરવો (Deleting the chart)

આલેખ દૂર કરવા (ડિલીટ કરવા) માટે સૌપ્રથમ તે પસંદ કરો અને Delete બટન દબાવો.

આલેખનું મુદ્રણ કરવું (Printing the chart)

અનેક સમયે તમારે આલેખ અને વર્કશીટની માહિતી છાપવાની જરૂર પડે છે. આલેખ છાપવા માટે નીચે જણાવેલાં સરળ પગલાં પ્રમાણે કાર્ય કરો :

- **File → Print** આદેશ આપો. આથી આકૃતિ 8.22માં દર્શાવ્યા પ્રમાણેનું ડાયલોગ બોક્સ તમે જોઈ શકશો.
- ઉપલબ્ધ પ્રિન્ટરની યાદીમાંથી પસંદગી કરો.
- હવે **OK** બટન દબાવો.



આકૃતિ 8.22 : દસ્તાવેજ છાપવો

અહીં એ નોંધ કરો કે કોઈ પણ દસ્તાવેજ છાપતાં પહેલા તે પાનું છાપ્યા પછી કેવું દેખાશે (પ્રિવ્યૂ preview) તે તપાસી લેવું ઘણું અગત્યનું છે. તે (પ્રિવ્યૂ) તમને તમારો દસ્તાવેજ કેવી રીતે છપાશે તેનું સ્પષ્ટ ચિત્ર આપશે.

અન્ય મલ્ટિમીડિયા ઓબ્જેક્ટ ઉમેરવા (Inserting other multimedia objects)

તમે વર્કશીટમાં આલેખ ઉપરાંત અન્ય કોઈ પણ મલ્ટિમીડિયા ઓબ્જેક્ટ ઉમેરી શકો છો. મલ્ટિમીડિયા ઓબ્જેક્ટમાં ચિત્રો, કેમેરાની ક્લિપ, વીડિયો, મૂવી ક્લિપ, એનિમેશનવાળી ક્લિપ જેમ કે GIF ફાઈલ અને 3D શાબ્દિક માહિતી જેવાં વિવિધ માધ્યમોનો સમાવેશ કરવામાં આવે છે. આ કાર્ય માટે આપણે સામાન્ય રીતે Insert જેવાં સરળ મેનુ આદેશનો ઉપયોગ કરીએ છીએ. એટલે કે, કેમેરા ક્લિપ કે તમારા મોબાઈલ ફોનની ક્લિપ ઉમેરવા માટે તમારે સૌપ્રથમ તમારા કમ્પ્યુટરને તે એકમ સાથે જોડવાની જરૂર પડે છે. તમે તાર સાથે જોડાણ (USB કનેક્ટર સાથેનો તાર (Cord) મોબાઈલ ફોન કે કેમેરા સાથે હોય તો) અથવા બ્લૂટૂથ ટેકનોલોજીનો ઉપયોગ કરી શકો છો. તમારા ફોટોગ્રાફ જેવો ઓબ્જેક્ટ તમારા કમ્પ્યુટરમાં હોય તો તે ઇચ્છિત ફોટોગ્રાફ ઉમેરવા માટે **Insert → Picture** આદેશ આપો. આ જ પ્રમાણે તમે કેલ્સી દસ્તાવેજમાં કોઈ પણ મૂવી કે એનિમેશનવાળી ક્લિપ (અથવા GIF ફાઈલ) ઉમેરી શકો છો.

તમે વર્કશીટમાં 3D શાબ્દિક માહિતી પણ ઉમેરી શકો છો. કેલ્સીમાં 3D શાબ્દિક લખાણ ગ્રાફિકલ ટેક્સ્ટ (graphical text) આર્ટ અથવા ફોન્ટવર્ક (fontwork) તરીકે ઓળખાય છે. આકૃતિ 8.23માં એક ઉદાહરણ આપેલ છે.



આકૃતિ 8.23 : કેલ્સીમાં ફોન્ટવર્ક

ફોન્ટવર્ક (fontwork) ઓબ્જેક્ટ બનાવવા માટે નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરો :

- Drawing ટૂલબાર કે **Fontwork** ટૂલબાર ઉપર રહેલા **Fontwork Gallery** આઈકોન ઉપર ક્લિક કરો.
- જો ડ્રોઈંગ ટૂલબાર અથવા ફોન્ટવર્ક ટૂલબાર સ્ક્રીન ઉપર દેખાતાં ન હોય તો **View** મેનુમાંથી **Toolbars** વિકલ્પ પસંદ કરો જેથી ટૂલબાર સ્ક્રીન ઉપર દેખાય (**View → Toolbars**).
- **Fontwork Gallery** ડાયલોગ બોક્સમાંથી ફોન્ટવર્ક સ્ટાઈલ નક્કી કરી OK બટન દબાવો. તમે ફોન્ટવર્ક ગેલેરીમાંથી સ્ટાઈલ (style) વાપરી શકો, Fontwork **shape** વડે તમે આકારમાં ફેરફાર કરી શકો, તમે અક્ષરોની ઊંચાઈ બદલી શકો, તમે શાબ્દિક લખાણની ગોઠવણ (alignment) કરી શકો અને શાબ્દિક લખાણમાં બે અક્ષરો વચ્ચેની જગ્યા સેટ કરી શકો. તમે રેખાનો રંગ, ભરેલો રંગ (fill colour) અને રંગ ભરવાની શૈલી (fill style) જેવાં વધારાનાં ગુણધર્મોમાં ફેરફાર કરવા માટે ડ્રોઈંગ ટૂલબારનો ઉપયોગ કરી શકો.
- તમારા દસ્તાવેજમાં ફોન્ટવર્ક ઓબ્જેક્ટને ઉમેરી શકો છો. તમારું પોતાનું લખાણ ઉમેરવા માટે ઓબ્જેક્ટ ઉપર ડબલ ક્લિક કરો.
- Edit મોડમાંથી બહાર નીકળવા માટે Esc દબાવો.

એક વાર ફોન્ટવર્ક ઓબ્જેક્ટ ઉમેરી દીધા પછી તેમાં સુધારા-વધારા કરવા માટે (edit) ફક્ત તેના ઉપર ક્લિક કરો. ફોન્ટવર્ક ટૂલબાર પ્રદર્શિત થશે. શાબ્દિક લખાણને સુધારો. તમારા સુધારા-વધારા કરવાનું કાર્ય પૂર્ણ થાય તે પછી Esc કી દબાવો.

સારાંશ

આ પ્રકરણમાં આપણે વિવિધ પ્રકારના આલેખો, તેની અગત્ય અને તેના ઉદ્દેશો વિશે શીખ્યા. આપણે કેલ્સી સ્પ્રેડશીટ પેકેજમાં આલેખ કઈ રીતે ઉમેરી શકાય તેમજ તેમાં કેવી રીતે ફેરફાર કરી શકાય તે બાબત પણ જોયું. આલેખના અનેક ભાગ હોય છે. આ પ્રકરણમાં આલેખના સામાન્ય ભાગની વિસ્તૃત સમજણ આપી. આપણે વધારે અસરકારક રજૂઆત કરવા માટે આલેખના ભાગને કઈ રીતે વધારે સારા બનાવી શકાય અને સુધારી શકાય તે બાબત પણ શીખ્યા. આ પ્રકરણમાં આપણે આલેખના બે પ્રકાર કોલમ ચાર્ટ અને પાઈ ચાર્ટ બાબત પણ ઊંડાણપૂર્વક ચર્ચા કરી. આપણે આલેખ સાથે હાઈપરલિંક કઈ રીતે જોડી શકીએ તે પણ જોયું. આ ઉપરાંત આ પ્રકરણમાં આલેખને છાપવો, આલેખની નકલ કરવી, આલેખ દૂર કરવો અને આલેખનો સુવાહ્ય દસ્તાવેજ (PDF) તરીકે નિકાસ કરવો તે બાબત પણ જોયું. અંતમાં 3D અને મલ્ટીમીડિયા ઓબ્જેક્ટ સ્પ્રેડશીટ દસ્તાવેજમાં કઈ રીતે ઉમેરી શકાય તે બાબત પણ ચર્ચા કરી.

સ્વાધ્યાય

1. કેલ્સીમાં આલેખ બનાવવાના ફાયદા સમજાવો.
2. આલેખના જુદા જુદા પ્રકારો વિશે એક એક લીટીમાં સમજૂતી આપો.
3. કેલ્સીમાં આલેખ કઈ રીતે ઉમેરી શકાય તે સમજાવો.
4. ચાર્ટ વિઝાર્ડના વિવિધ ભાગની યાદી બનાવો.
5. આલેખના વિવિધ ભાગની યાદી સાથે દરેકનું એક એક લીટીમાં વર્ણન કરો.
6. એક વખત આલેખ તૈયાર થઈ જાય પછી તેનો પ્રકાર કઈ રીતે બદલી શકાય ?
7. આલેખના અક્ષરને ફોર્મેટ કઈ રીતે કરી શકાય ?
8. આલેખ સાથે હાઈપરલિંક જોડી શકાય ? કઈ રીતે ?
9. નીચેનામાંથી સાચો વિકલ્પ પસંદ કરો :
 - (1) આલેખ છાપવા માટે નીચેનામાંથી કયો વિકલ્પ વપરાય છે ?
 - (a) Insert → Chart
 - (b) File → View
 - (c) File → Print
 - (d) View → Chart
 - (2) કેલ્સીમાં બનાવેલા આલેખમાં કેટલા અક્ષ હોઈ શકે ?
 - (a) બે
 - (b) ત્રણ
 - (c) બે અથવા ત્રણ
 - (d) ચાર
 - (3) કેલ્સીમાં 3D શાબ્દિક લખાણ દાખલ કરવા માટે નીચેનામાંથી શું વપરાય છે ?
 - (a) ફોન્ટવર્ક
 - (b) આર્ટવર્ક
 - (c) ડ્રોઈંગવર્ક
 - (d) ગ્રાફવર્ક
 - (4) આલેખને જોડવા માટે નીચેનામાંથી શું વપરાય છે ?
 - (a) એક્ટિવલિંક
 - (b) હાઈપરલિંક
 - (c) ડ્રોઈંગલિંક
 - (d) કનેક્શન લિંક
 - (5) નીચેનામાંથી કયો વિકલ્પ આલેખનો પ્રિવ્યૂ દેખાડે છે ?
 - (a) XHTML તરીકે સંગ્રહ
 - (b) પેઈજ પ્રિવ્યૂ
 - (c) આલેખની નિકાસ
 - (d) આપેલ કોઈ પણ નહીં
 - (6) ડેટા વિસ્તારમાં ઊભી હરોળ (કોલમ) અથવા આડી હરોળ(રો)ને અચળ બનાવવા માટે - નિરપેક્ષ બનાવવા માટે કયું ચિહ્ન વપરાય છે ?
 - (a) #
 - (b) \$
 - (c) &
 - (d) %
 - (7) નીચેનામાંથી શેની સાથે કેલ્સી આલેખને જોડી શકાય છે ?
 - (a) હયાત દસ્તાવેજ સાથે
 - (b) નવા દસ્તાવેજ સાથે
 - (c) વેબ પેઈજ સાથે
 - (d) આપેલા તમામ વિકલ્પો

પ્રાયોગિક સ્વાધ્યાય

1. અંગ્રેજી, વિજ્ઞાન, ગણિત વગેરે વિષયોમાંથી તમારા ગુણ શોધો. દરેક વિષયમાં વર્ગના સૌથી વધારે ગુણ શોધો. દરેક વિષયના મહત્તમ ગુણ સાથે તમારા ગુણની સરખામણી કરતો કોલમ ચાર્ટ બનાવો.
2. તમારા પડોશમાં દરેક વ્યક્તિ ક્યું છાપું વાંચે છે તેનું એક નાનું સર્વેક્ષણ કરો. ડેટા ભેગો કરો, કોરા પત્રક ઉપર ગોઠવણી કરો અને તેના આધારે એક 3D પાઈ ચાર્ટ બનાવો.
3. કોઈ એક ગાણિતિક પદાવલિ જેમ કે $Y = 2 \cdot X$ લો. નીચે જણાવ્યા પ્રમાણે Xની જુદી જુદી કિંમત લો :

Value of X	Value of Y
1	?
3	?
5	?
8	?
6	?

નીચે જણાવ્યા પ્રમાણે કરો :

- (a) કેલ્સી વર્કશીટમાં કિંમત દાખલ કરો.
 - (b) સૂત્ર વાપરીને Xની દરેક કિંમત માટે Yની કિંમત શોધો.
 - (c) X અને Y કિંમતો સાથે લઈને લાઈન ચાર્ટ બનાવો.
 - (d) તમને ઈચ્છા થાય તે પ્રમાણે આલેખને ફોર્મેટ કરો.
4. નીચે જણાવેલ કોઈ પરિવારનો અંદાજપત્ર સંબંધિત ડેટા લો :

Category	July
Provision	1000
Milk	500
Vegetables	500
Servant	800
Electricity	1500
Gas	500
Fees	1500
Newspaper	100
Cable	200
Mobile	200

નીચે જણાવ્યા પ્રમાણે કરો :

- (a) આ ડેટા સ્પ્રેડશીટમાં દાખલ કરો અને પાઈ ચાર્ટ બનાવો.
- (b) આલેખ તૈયાર થઈ ગયા પછી તમારી પસંદ પ્રમાણે ફોર્મેટ કરો. દા.ત. કોઈ ભાગનો રંગ બદલો, ફોન્ટની શૈલી બદલો, લિજેન્ડ ઉમેરો / દૂર કરો, આલેખમાં ડેટા લેબલ ઉમેરો, આલેખમાંથી થોડા ખંડ બહાર કાઢો.
- (c) XHTML ફાઈલ તરીકે આલેખની નિકાસ કરો. વેબ બ્રાઉઝર ખોલો અને જુઓ કે તે આલેખ કેવો દેખાય છે.



સમસ્યા અને સમસ્યાનું નિરાકરણ

દરેક મનુષ્ય પોતાની રોજિંદી જિંદગીમાં અનેક સમસ્યાઓનું નિરાકરણ લાવે છે. ચાલો, આપણે આવી એક ક્રિયાની ચર્ચા કરીએ. ધારો કે તમે તમારા મિત્રો સાથે સંતાકુકડી રમત (Hide and seek) રમો છો. આ રમતમાં આપણી સમસ્યા એ છે કે આપણા છુપાયેલા મિત્રની જગ્યા શોધવાની છે. આ કાર્ય ફક્ત તો જ થઈ શકે જો આપણે તેની છુપાવાની જગ્યા વિશે આગાહી કરી શકીએ.

ચાલો, આપણે બીજી એક સમસ્યા લઈએ, ધારો કે આપણે એક ગાણિતિક સમીકરણનો ઉકેલ લાવવો છે જેમાં આપણને $2x + 4 = 0$ સમીકરણ આપેલું છે તો x ની શું કિંમત હોઈ શકે ? આ સમસ્યાના જવાબ માટે આપણે સમીકરણને $2x = -4$ ની રીતે ગોઠવણી કરવી પડશે, અને પછી $x = -4/2$, આથી અહીં જવાબ -2 મળશે.

ચાલો, આપણે અન્ય ઉદાહરણ લઈએ. આપણે અંગ્રેજી શબ્દ "Eloquent" નો અર્થ શોધવો છે. આ માટે આપણે એક અંગ્રેજી ભાષાનો શબ્દકોશ જોઈએ. શબ્દકોશમાં હજારો શબ્દો અને તેના અર્થ હોય છે. છતાં આપણે ઘણી ઝડપથી ઈચ્છિત શબ્દનો અર્થ શોધી કાઢીએ છીએ. અત્યાર સુધીમાં ક્યારેય પણ તમે વિચાર્યું છે કે આ કાર્ય તમે આટલી ઝડપથી કઈ રીતે કરી શકો છો ? હકીકતમાં શબ્દકોશમાં શબ્દો ક્રમિક રીતે ગોઠવાયેલા હોય છે તેનો આપણે ફાયદો લઈએ છીએ. આપણે જે શબ્દો "E" થી શરૂ ન થતા હોય તેને ઝડપથી કાઢી નાખીએ છીએ અને "E" થી શરૂ થાય તેવા શબ્દો શોધવાનું શરૂ કરીએ છીએ, એ જ રીતે બીજો અક્ષર "I" હોય તેવા શબ્દો શોધવા માટે કાઢી નાખવાની પદ્ધતિ (elimination method) નો ઉપયોગ કરીએ છીએ. આપણે આ કાઢી નાખવાની પદ્ધતિનો જ ઉપયોગ બાકીના અક્ષરો શોધવામાં ચાલુ રાખીએ છીએ કે જ્યાં સુધી આપણો ઈચ્છિત શબ્દ મળી ન જાય.

ઉપરના ઉદાહરણથી આપણને ખ્યાલ આવ્યો કે સમસ્યા કેવી હોઈ શકે ? સમસ્યાઓનું નિરાકરણ માનસિક શક્તિઓ અથવા યંત્રોના કાર્ય (કોઈ પદ્ધતિસરના કાર્ય) વપરાશકર્તાનું ઈચ્છિત પરિણામ મેળવવા તરફ લઈ જાય છે. ઉપર ચર્ચા કરેલાં ઉદાહરણો દર્શાવે છે કે કેટલીક સમસ્યાઓના પરિણામની આગાહી ચોક્કસ રીતે કરી શકાય છે જ્યારે કેટલીક સમસ્યાઓના ચોક્કસ પરિણામની આગાહી કરવી ઘણી મુશ્કેલ છે. આ રીતે સમસ્યાઓનું બે પ્રકારમાં વર્ગીકરણ કરી શકાય : સારી રીતે વ્યાખ્યાયિત કરેલી અને સ્પષ્ટ રૂપરેખા વગરની સમસ્યાઓ. ઉપર જણાવેલાં ઉદાહરણોમાં બીજો અને ત્રીજો દાખલો એ સ્પષ્ટ રીતે વર્ણવેલી સમસ્યાનો છે જ્યારે પહેલો દાખલો એ સ્પષ્ટ રૂપરેખા વગરનો છે. અહીં તમે નોંધ કરશો કે સ્પષ્ટ સમસ્યાઓનાં લક્ષ્ય ચોક્કસ અને સ્પષ્ટ હોય છે અને આથી આપણે સમસ્યાના ઉકેલનાં તબક્કા (પગલાં) પણ સ્પષ્ટ રીતે વર્ણવી શકીએ છીએ. એ પણ હકીકત છે કે કમ્પ્યુટર ફક્ત સ્પષ્ટ રૂપરેખા ધરાવતી સમસ્યાઓનો જ ઉકેલ લાવે છે અને આથી આપણે આ પ્રકરણમાં ફક્ત સ્પષ્ટ રૂપરેખા ધરાવતી (well defined) સમસ્યાઓ બાબત જ ચર્ચા કરીશું.

કમ્પ્યુટરનાં ક્ષેત્રમાં કોઈ પણ આપેલી સમસ્યાઓનો ઉકેલ એ કમ્પ્યુટરને આપેલી ક્રમિક સૂચનાઓ જ છે. કમ્પ્યુટર અતિ સરળતાથી ઘણી ગૂંચવાડાભરી વિવિધ પ્રકારની સમસ્યાઓને ઉકેલે છે. સમસ્યાના ઉકેલ માટે કમ્પ્યુટરને સૂચનાઓનો સંપૂર્ણ ગણ આપવો પડે છે. આ સૂચનાઓ કમ્પ્યુટરને દરેક તબક્કે શું કરવાનું છે તે જણાવે છે. એક વસ્તુ ખાસ યાદ રાખો કે કમ્પ્યુટર જાતે સમસ્યાનો ઉકેલ લાવી શકતું નથી પણ તે સમસ્યાના ઉકેલમાં ફક્ત મદદરૂપ થાય છે. નીચે જણાવેલાં પગલાંને અનુસરવાથી આપણે કોઈ પણ સમસ્યાનો ઉકેલ લાવી શકીએ છીએ :

1. સમસ્યાને વ્યાખ્યાયિત કરો.
2. ઈનપુટ, આઉટપુટ અને સમસ્યાના અવરોધોને ઓળખો.
3. સમસ્યાના ઉકેલના અલગ અલગ વિકલ્પો શોધો.
4. ઉપર જણાવેલા વિવિધ વિકલ્પોની યાદીમાંથી સારામાં સારો વિકલ્પ પસંદ કરો.
5. આપણે નક્કી કરેલા વિકલ્પ માટે વિગતવાર ક્રમિક સૂચનાઓ તૈયાર કરો.
6. આપણે તૈયાર કરેલા સૂચનાઓના ગણ દ્વારા પરિણામની ગણતરી કરો.
7. આપણે મેળવેલો જવાબ સાચો છે કે કેમ તે ચકાસો.

જે વ્યક્તિએ ઉકેલ મેળવવો છે તેણે પગલાં 1થી 5 કરવાનાં છે જ્યારે પગલાં 6 અને 7 કમ્પ્યુટર વડે થશે.

ધારો કે આપણે કોઈ આપેલી સંખ્યા એકી છે કે બેકી તે શોધવું છે. આ પ્રશ્નના ઉત્તર માટે નીચે આપેલી સૂચનાઓનો ઉપયોગ કરી શકાય :

1. સંખ્યાનું ઇનપુટ લો.
2. તે સંખ્યાને 2 વડે ભાગો અને શેષ શોધો.
3. જો શેષ 1 રહે તો આપેલી સંખ્યા એકી છે નહીંતર તે સંખ્યા બેકી છે.

આ સમસ્યાનો સર્વસામાન્ય ઉકેલ નીચે જણાવેલી ત્રણ રીતો વડે મેળવી શકાય :

1. સુડો કોડ (Pseudo code) - અનુકરણ કોડ
2. ફ્લોચાર્ટ (Flowchart) - કમ્પ્યુટર રેખાચિત્ર
3. અલ્ગોરિથમ (Algorithm) - કલનવિધિ

આપેલી સંખ્યા એકી છે કે બેકી તે જાણવા માટે ઉપર જણાવેલાં ત્રણ પગલાં એ સુડો કોડ (Pseudo code) છે. સુડો એટલે આભાસી અથવા અનુકરણ કરેલો (simulated). આપણે આમાંનો બીજો અર્થ અહીં સ્વીકારીશું કારણ કે અહીં આપણા સંદર્ભમાં તે વધારે બંધબેસતો શબ્દ છે. આપણે આ પ્રશ્નના ઉકેલ માટે ઉપર જણાવેલાં ત્રણ પગલાંને અનુકરણ કરેલો કોડ કહી શકીએ. ચાલો, હવે અન્ય બે રીતો વિશે પણ જાણીએ.

ફ્લોચાર્ટ (Flowchart)

ફ્લોચાર્ટ એ એક કાર્યરીતિ છે જેમાં સમસ્યાના ઉકેલ માટે યંત્ર દ્વારા કરવામાં આવતા દરેક કાર્યને ચિત્રાત્મક સ્વરૂપે રજૂ કરવામાં આવે છે. જુદા જુદા કાર્યો માટે વિવિધ ચિહ્નોનો ઉપયોગ ફ્લોચાર્ટ બનાવવામાં થાય છે. આ ચિહ્નોને ફ્લોચાર્ટનાં ઘટકો કહેવામાં આવે છે. આપણી પાસે પ્રક્રિયાના દરેક કાર્ય માટે અલગ અલગ (અજોડ) ચિહ્ન છે. ચાલો, હવે આપણે વધારે વપરાતા ઘટકો અને તેનાં ચિહ્નો બાબત ચર્ચા કરીએ.

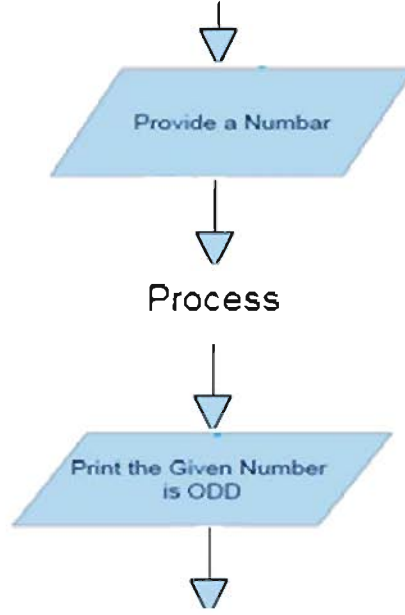
આરંભ અને અંત (Start and End) : આરંભ અને અંત ઘટકો ફ્લોચાર્ટની શરૂઆત અને અંત બતાવવા માટે વપરાય છે. તે આકૃતિ 9.1માં દર્શાવ્યા પ્રમાણે અંડાકાર હોય છે. આ ચિહ્નોને અંતિમ ચિહ્નો (ટર્મિનલ સિમ્બોલ) પણ કહેવામાં આવે છે.



આકૃતિ 9.1 : ટર્મિનલ સિમ્બોલ

કોઈ પણ ફ્લોચાર્ટમાં આ ચિહ્ન બે વાર વપરાય છે : ફ્લોચાર્ટની શરૂઆતમાં તેમજ તેના અંતમાં.

ઇનપુટ - આઉટપુટ (Input-Output) : દરેક સમસ્યાને ઇનપુટ (નિવેશ)ની જરૂર હોય છે, આ નિવેશ ઉપર પ્રક્રિયા કરવામાં આવે છે અને તે આઉટપુટ (નીપજ-ઉત્પાદન) તૈયાર કરે છે. ફરી ઉપર જણાવેલો પ્રશ્ન 3 લો, અહીં આપણે એક સંખ્યાના ઇનપુટની જરૂર છે, આ આપેલી સંખ્યાને 2 વડે ભાગવાની અને શેષ શોધવાની એ પ્રક્રિયા છે અને આપણે આપેલી સંખ્યા એકી છે કે બેકી તે નિર્ણય એ આપણું આઉટપુટ છે. આથી આપણે એક ઇનપુટ અને એક આઉટપુટ દર્શાવવા માટે ચિહ્ન જોઈએ. ફ્લોચાર્ટમાં ઇનપુટ અને આઉટપુટ દર્શાવવા માટે આકૃતિ 9.2માં દર્શાવ્યા પ્રમાણે સમાંતર ચતુર્ભુજ વપરાય છે.

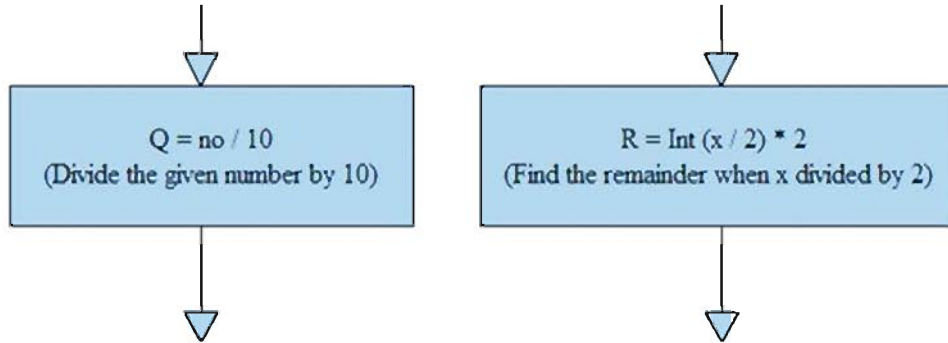


આકૃતિ 9.2 : ઇનપુટ-આઉટપુટ માટેનું ચિહ્ન

તીર (એરો - Arrow) : આકૃતિ 9.1 અને આકૃતિ 9.2નું નિરીક્ષણ કરો તો જણાશે કે આપણે ચિહ્નની અંદર જતું તેમજ ચિહ્નમાંથી બહાર આવતું તીર (એરો) બતાવેલું છે. કાર્ય જે ક્રમમાં કરવાનું હોય તે દર્શાવવા માટે તીરનો ઉપયોગ થાય છે. સામાન્ય રીતે તે ચિહ્નથી શરૂ થાય છે અને બીજા ચિહ્ન ઉપર તેનો અંત આવે છે. આ રીતે એક તીર એક ચિહ્નમાંથી નીકળે છે અને એક તીર તે ચિહ્નમાં અંદર જાય છે.

અહીં નોંધ કરશો કે start ચિહ્નમાંથી ફક્ત બહાર જતું તીર છે જ્યારે End ચિહ્નમાં ફક્ત તેની અંદર જતું તીર છે.

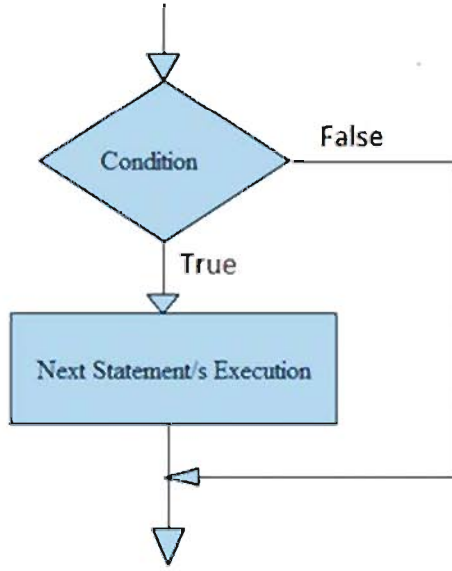
પ્રક્રિયા (Process) : કોઈ પણ સમસ્યાના ઉકેલમાં પ્રક્રિયા (process) એ મુખ્ય ભાગ છે. હકીકતમાં પ્રક્રિયા એ ક્રિયાઓની એક શ્રેણી છે. પ્રક્રિયાને દર્શાવવા માટે આકૃતિ 9.3માં દર્શાવ્યા પ્રમાણે એક લંબચોરસનું ચિહ્નનો આપણે ઉપયોગ કરીએ છીએ.



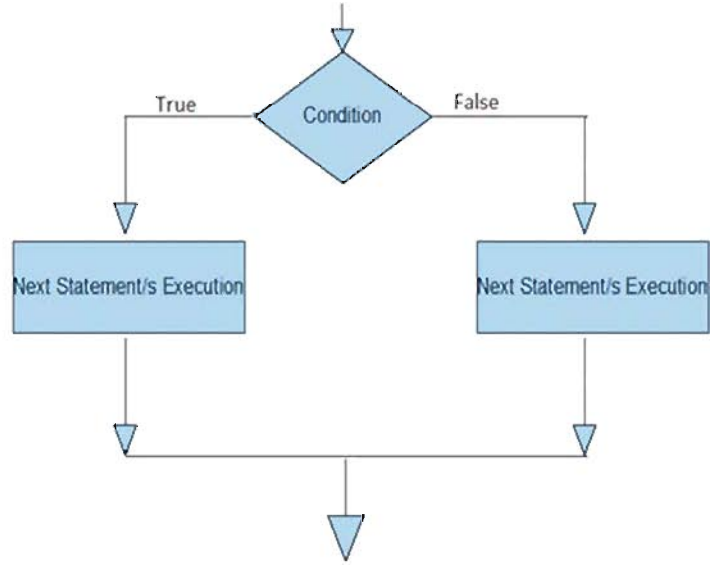
આકૃતિ 9.3 : પ્રક્રિયાનું ચિહ્ન

સામાન્ય રીતે કમ્પ્યુટરની પ્રક્રિયા એ અંગકગણિતનું કોઈ કાર્ય અથવા તાર્કિક ક્રિયા હોય છે. અંગકગણિતનું કાર્ય એટલે સરવાળા, બાદબાકી, ગુણાકાર અથવા ભાગાકાર. તાર્કિક કાર્ય સામાન્ય રીતે નિર્ણય લેવામાં મદદરૂપ થાય છે જ્યારે તે કોઈ પ્રશ્નના હા કે ના પ્રકારના જવાબ માટે વપરાય છે. દાખલા તરીકે, આપણે પ્રશ્ન પુછીએ કે 10 એ 5 કરતાં મોટો છે ? આ પ્રશ્નનો જવાબ 'હા' અથવા 'ના' હોય છે.

નિર્ણય (Decision) : તાર્કિક નિર્ણયની પ્રક્રિયાને આકૃતિ 9.4માં દર્શાવ્યા પ્રમાણે એક હીરા આકારનાં ચિહ્નથી દર્શાવવામાં આવે છે. તેને પરીક્ષણ ચિહ્ન (ટેસ્ટ સિમ્બોલ) પણ કહેવામાં આવે છે. જ્યારે આપણે ઉકેલ માટેની સામાન્ય શ્રેણી બદલવા ઇચ્છતા હોય (જુઓ આકૃતિ 9.4) અથવા નિર્ણયનાં પરિણામનાં આધારે કોઈ ચોક્કસ વિધાનનો અમલ કરવાની જરૂર હોય (જુઓ આકૃતિ 9.5) ત્યારે નિર્ણયપેટી (decision box) વપરાય છે.

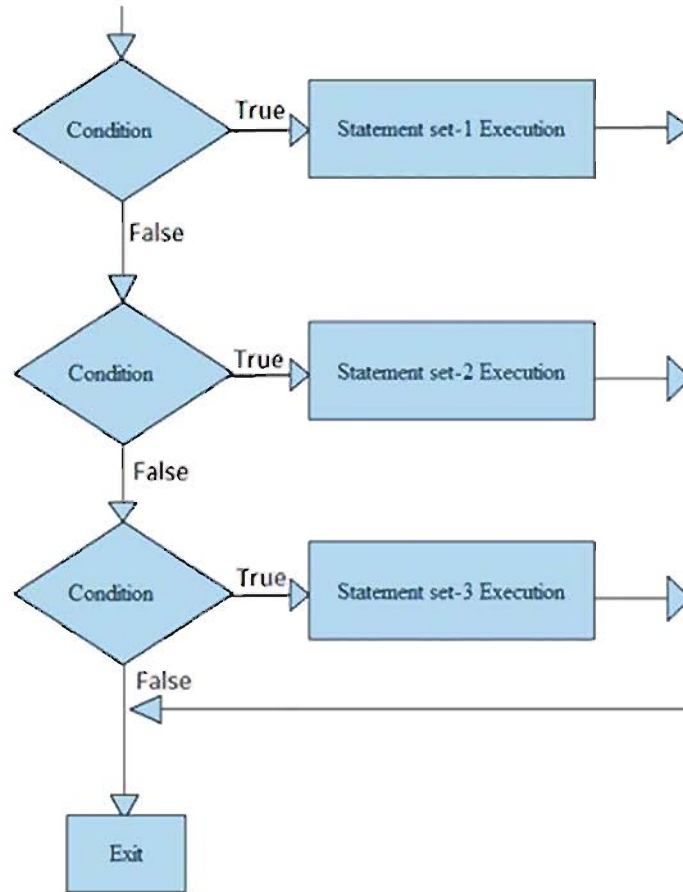


આકૃતિ 9.4 : નિર્ણય ચિહ્ન



આકૃતિ 9.5 : જુદાં જુદાં પરિણામોનો અમલ

અમુક સમયે નિર્ણય લેવામાં બે કરતાં વધારે વિકલ્પોની આપણને જરૂર પડે છે. આવા સમયે આકૃતિ 9.6માં દર્શાવ્યા પ્રમાણે એક કરતાં વધારે નિર્ણયપેટીઓને જોડીને જરૂરી વિકલ્પ બનાવવામાં આવે છે.

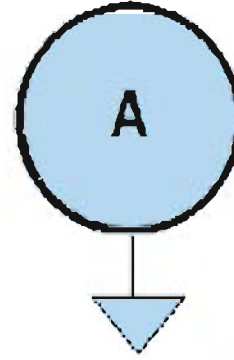
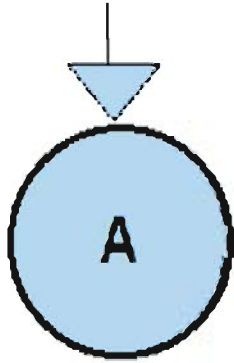


આકૃતિ 9.6 : એક કરતાં વધારે નિર્ણયોનું જોડાણ

આકૃતિ 9.6માં દર્શાવ્યા પ્રમાણે સૌપ્રથમ શરત તપાસવામાં આવે છે અને જો તેનું પરિણામ ખરું આવે તો પહેલાં વિધાનોના ગણનો અમલ થશે. જો પરિણામ ખોટું આવે તો બીજી શરત તપાસવામાં આવશે. આ પ્રક્રિયા આપેલી બધી શરતો માટે ફરીફરીને કરવામાં આવશે. અહીં એ નોંધ કરો કે જ્યારે કોઈ એક શરતનું પરિણામ ખરું આવે તો તે પછીની શરત તપાસવામાં આવતી નથી.

અનુસંધાન ચિહ્ન Connector : અનુસંધાન ચિહ્ન દર્શાવવા માટે વર્તુળનું ચિહ્ન વપરાય છે. અમુક સમયે એ શક્ય છે કે ફ્લોચાર્ટ એટલો મોટો બની જાય કે તે એક પાનામાં સમાવી ન શકાય અથવા બે પ્રક્રિયાને તીર વડે જોડવી શક્ય ન બને. આવી પરિસ્થિતિમાં બે ભાગને જોડવા માટે અનુસંધાન ચિહ્ન (કનેક્ટર) વાપરવામાં આવે છે. જોડાણ માટે ઓછામાં ઓછા બે વર્તુળ જરૂરી છે; આકૃતિ 9.7માં દર્શાવ્યા પ્રમાણે એક તીર વર્તુળની અંદર જતું અને બીજું તીર વર્તુળમાંથી બહાર આવતું. જુદા જુદા અનુસંધાન ચિહ્ન ઓળખવા માટે અલગ અલગ અક્ષરો જોડીમાં વપરાય છે.

Break From One Flowchart



Joining Another Flowchart

આકૃતિ 9.7 : અનુસંધાન ચિહ્ન (કનેક્ટર)

ફ્લોચાર્ટનાં ઉદાહરણો :

અત્યાર સુધીમાં આપણે જાણ્યું કે ફ્લોચાર્ટ શું છે અને તેમાં વપરાતા ચિહ્નો બાબત જાણકારી મેળવી. ચાલો, હવે આપણે ફ્લોચાર્ટ બનાવીને કેટલીક સમસ્યાઓનો ઉકેલ લાવીએ.

ઉદાહરણ 1 : જમીન ઉપર લાદી લગાવવાની કિંમત 10 રૂ. ચો. ફીટ છે. હવે ધારો કે આપણી સમસ્યા લંબચોરસ આકારના નગરના સભાગૃહમાં જમીન ઉપર લાદી લગાવવાની કુલ કિંમત શોધવાની છે.

જવાબ :

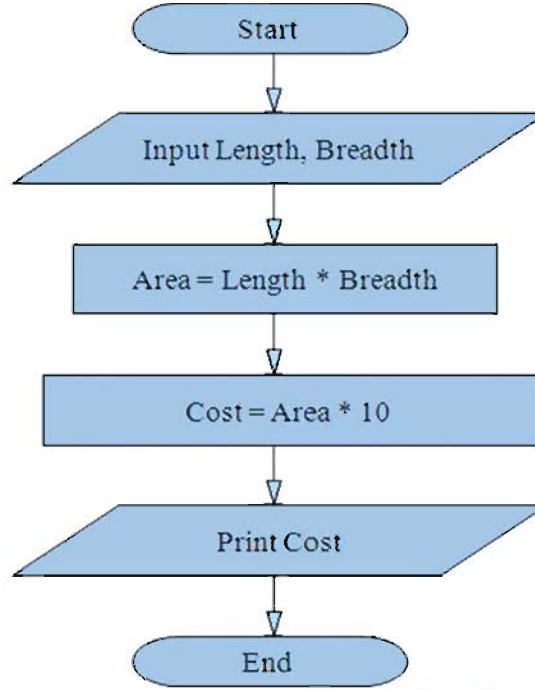
જમીન ઉપર લાદી લગાવવાનો કુલ ખર્ચ શોધવા માટે આપણે પહેલાં જમીનનું ક્ષેત્રફળ શોધવું પડે. આપણે જાણીએ છીએ કે લંબચોરસનું ક્ષેત્રફળ એ તેની લંબાઈ અને પહોળાઈનો ગુણાકાર છે. આથી આપણે જો સભાગૃહની લંબાઈ અને પહોળાઈ જાણતા હોઈએ તો લંબચોરસનું ક્ષેત્રફળ શોધી શકીએ. જમીનનું ક્ષેત્રફળ શોધવા માટે આ મુજબ સૂત્ર આપી શકાય :

હવે જમીન ઉપર લાદી લગાવવાની કુલ કિંમત શોધવા માટે આપણે જમીનનું ક્ષેત્રફળ અને એક ચો. ફીટ લગાવવાની કિંમતનો ગુણાકાર કરવો પડે. કુલ કિંમત શોધવાનું સૂત્ર : કુલ કિંમત = ક્ષેત્રફળ × 10.

અહીં સમસ્યાના ઉકેલ માટે આપણે ચાર ચલ ક્ષેત્રફળ, લંબાઈ, પહોળાઈ અને કિંમતની જરૂર પડે છે. હજી આગળ વધતાં પહેલાં આપણે પહેલા શબ્દ ચલની વ્યાખ્યા કરીએ, કારણ કે તે બધા ફ્લોચાર્ટમાં વપરાય છે.

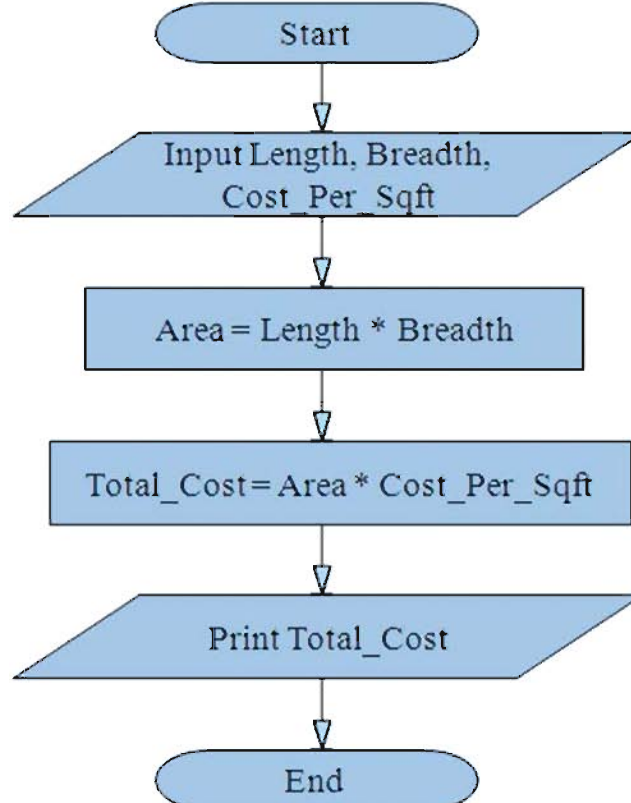
ચલ (Variable) : ચલ એ એક એન્ટિટી (entity) છે જેની કિંમત પ્રક્રિયાના અમલ દરમિયાન બદલી શકાય છે અને તેને કિંમત આપી શકાય છે.

જમીન ઉપર લાદી લગાવવાની કિંમત શોધવાનો ફ્લોચાર્ટ આકૃતિ 9.8માં દર્શાવ્યો છે.



આકૃતિ 9.8 : લાદી લગાવવાની કિંમત શોધવાનો ફ્લોચાર્ટ

પદાવલિ કિંમત = ક્ષેત્રફળ \times 10માં કિંમત 10 જુઓ, અહીં 10 એ અચળ (કિંમત) કહેવાય છે. આ એક એવી એન્ટિટી છે, જેની કિંમત એક વાર આપીએ તે પછી આખી પ્રક્રિયા દરમિયાન તેમાં ફેરફાર થતો નથી અને તે સ્થાયી રહે છે. આકૃતિ 9.8માં દર્શાવેલા ફ્લોચાર્ટ દ્વારા આપણે જે ઉકેલ મેળવીએ છીએ તે ફક્ત એ કિસ્સામાં જ વાપરી શકાય જ્યારે એક ચો. ફીટ લાદી લગાવવાની કિંમત રૂ. 10 હોય. જો લાદી લગાવવાની કિંમતમાં ફેરફાર થાય તો ઉપર જણાવેલો ઉકેલ ન ચાલે. ઉપર જણાવેલી સમસ્યાનો સર્વસામાન્ય ઉકેલ આકૃતિ 9.9માં આપેલો છે.



આકૃતિ 9.9 : લાદી લગાવવાનો ખર્ચ શોધવાનો સુધારાયેલ ફ્લોચાર્ટ

આકૃતિ 9.9નું નિરીક્ષણ કરો, અહીં આપણે ચારને બદલે પાંચ ચલનો ઉપયોગ કર્યો છે. અહીં Cost_Per_Sqft ચલનો ઉપયોગ એક ચો. ફીટ લાદી લગાવવાની કિંમતનો સંગ્રહ કરવા માટે કર્યો છે. દરેક સમયે આપણે તેની કિંમતમાં ફેરફાર કરી શકીએ અને આપણે જરૂરી આઉટપુટ મેળવી શકીએ. કોષ્ટક 9.1માં આવાં કેટલાક આઉટપુટ આપેલાં છે.

Length	Breadth	Area=Length*Breadth	Cost_Per_Sqft	Total_Cost=Area*Cost_Per_Sqft
50	50	2500	10	25000
25	75	1875	10	18750
45	35	1575	20	31500

કોષ્ટક 9.1 : આકૃતિ 9.9માં દર્શાવેલા ફ્લોચાર્ટનું આઉટપુટ

ઉદાહરણ 2 : ધારો કે તમારા શહેરનાં રમતગમત સંકુલમાં એક સુંદર અને વર્તુળ ક્રિકેટનું મેદાન છે. તેના સત્તાધીશો આ મેદાનની ફરતે વાડ બનાવવા ઇચ્છે છે. આ ઉપરાંત તેઓ સંપૂર્ણ મેદાનને લોન (વ્યવસ્થિત રીતે કાપેલા અને સુંવાળા ઘાસવાળી હરિયાળી જમીન)થી ઢાંકવા ઇચ્છે છે. સત્તાધીશો જાણવા માગે છે કે કેટલા મીટરની વાડ જોઈશે ? તેઓ એ પણ જાણવા ઇચ્છે છે કે મેદાનનું કુલ ક્ષેત્રફળ કેટલું છે કે જેને લોન વડે ઢાંકવાની જરૂર છે.

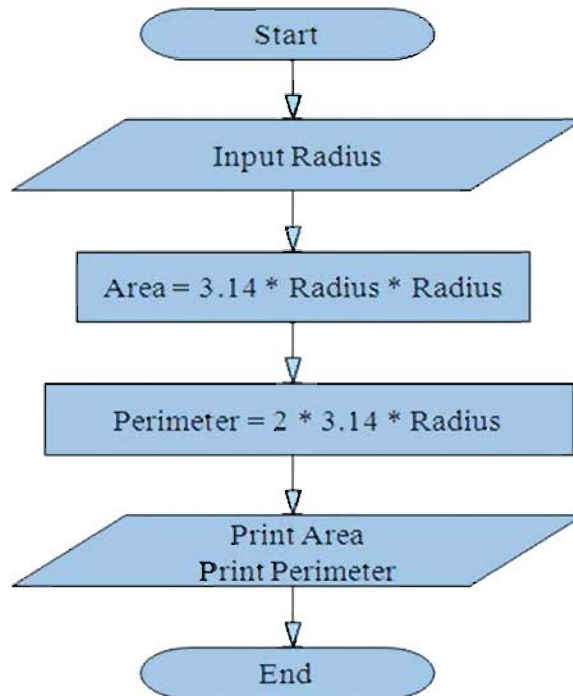
જવાબ :

આ પ્રશ્નના ઉકેલ માટે આપણે મેદાનનું ક્ષેત્રફળ અને તેની પરિમિતિ શોધવી પડશે. ગોળ મેદાનનું ક્ષેત્રફળ શોધવા માટેનું સૂત્ર ક્ષેત્રફળ = $\pi * R^2$ અને મેદાનની પરિમિતિ = $2 * \pi * R$ (અહીં π ની કિંમત 3.14 અને R એ મેદાનની ત્રિજ્યા છે)

આથી, આ પ્રશ્નના ઉકેલ માટે ત્રણ ચલ Radius (ત્રિજ્યા), Area (ક્ષેત્રફળ) અને Perimeter (પરિમિતિ) આપણને જરૂર પડશે. ફ્લોચાર્ટમાં આપણે ક્ષેત્રફળની ગણતરી અને મેદાનની પરિમિતિ શોધવા માટે જે સૂત્ર વાપરીશું તે નીચે આપેલાં છે.

Area = 3.14 * Radius * Radius (ક્ષેત્રફળ = 3.14 * ત્રિજ્યા * ત્રિજ્યા)

અને Perimeter = 2 * 3.14 * Radius (પરિમિતિ = 2 * 3.14 * ત્રિજ્યા)

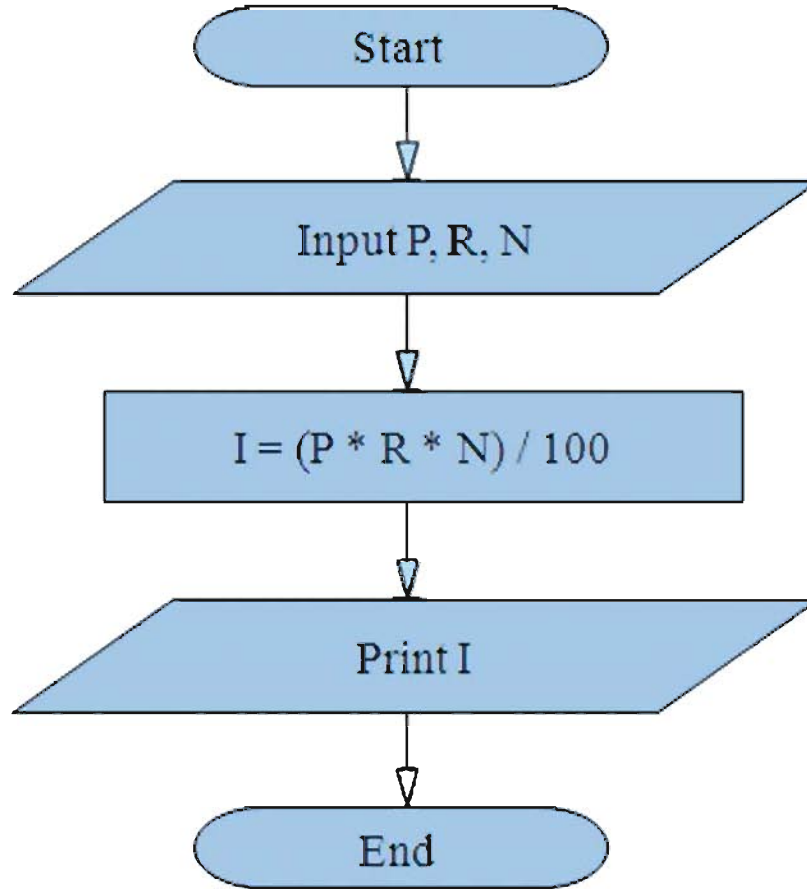


આકૃતિ 9.10 : વર્તુળનું ક્ષેત્રફળ અને પરિમિતિ શોધવાનો ફ્લોચાર્ટ

ઉદાહરણ 3 : જહાનવીએ બેન્કમાંથી 6 વર્ષની 11.5 ટકાના દરથી 35,000 રૂ.ની લોન લીધી. તેણીએ બેન્કને કેટલું સાદું વ્યાજ ચૂકવવું પડશે તેની ગણતરી કરો.

જવાબ

કોઈ લોનની મુદ્દલ રકમ 'P', વ્યાજનો દર 'R' અને 'N' વર્ષ માટેના સાદા વ્યાજ 'I' ની ગણતરી માટેનું સૂત્ર $I = (P * R * N) / 100$ છે. આ પ્રશ્નના ઉકેલ માટે આપણે ચાર ચલ I, P, R, N અને અચળ કિંમત 100ની જરૂર પડશે. આ પ્રશ્નનો ઉકેલ આકૃતિ 9.11માં ફ્લોચાર્ટના રૂપમાં દર્શાવ્યો છે.



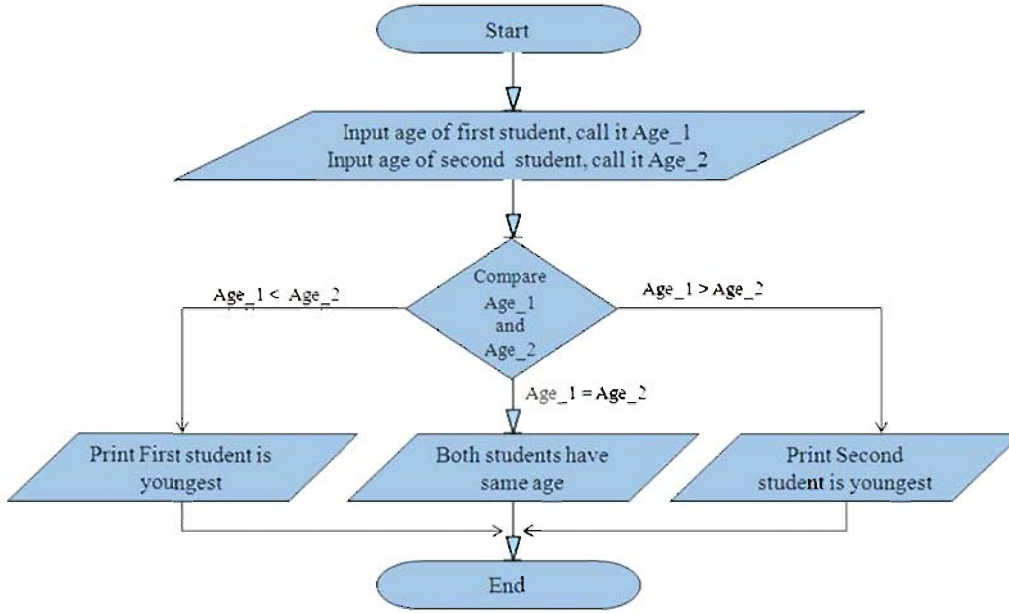
આકૃતિ 9.11 : સાદા વ્યાજની ગણતરી માટેનો ફ્લોચાર્ટ

ઉદાહરણ 4 : ધારો કે બે વિદ્યાર્થીઓમાંથી તમારે નાની ઉંમરનો વિદ્યાર્થી શોધવો છે, આ કિસ્સામાં વિદ્યાર્થીની ઉંમર એ નિવેશ (ઇનપુટ) હશે.

જવાબ :

ઉપરના પ્રશ્નનો ઉકેલ નીચે જણાવેલી ક્રિયાઓ કરવાથી મળશે :

સૌપ્રથમ બંને વિદ્યાર્થીઓની ઉંમરનું ઇનપુટ (નિવેશ) લો. તે ઉંમર Age_1 અને Age_2 ચલમાં રાખો. હવે Age_1 અને Age_2ની કિંમતની સરખામણી કરો. જો બંને કિંમત સરખી હોય તો આપણી પાસે સરખી ઉંમરના બંને વિદ્યાર્થી છે અને આથી બંને નાનામાં નાના છે. જો Age_1ની કિંમત Age_2 કરતાં ઓછી હોય તો પહેલો વિદ્યાર્થી નાનો છે, નહીંતર બીજો વિદ્યાર્થી. આ પ્રશ્નના ઉકેલનો ફ્લોચાર્ટ આકૃતિ 9.12માં આપેલો છે.



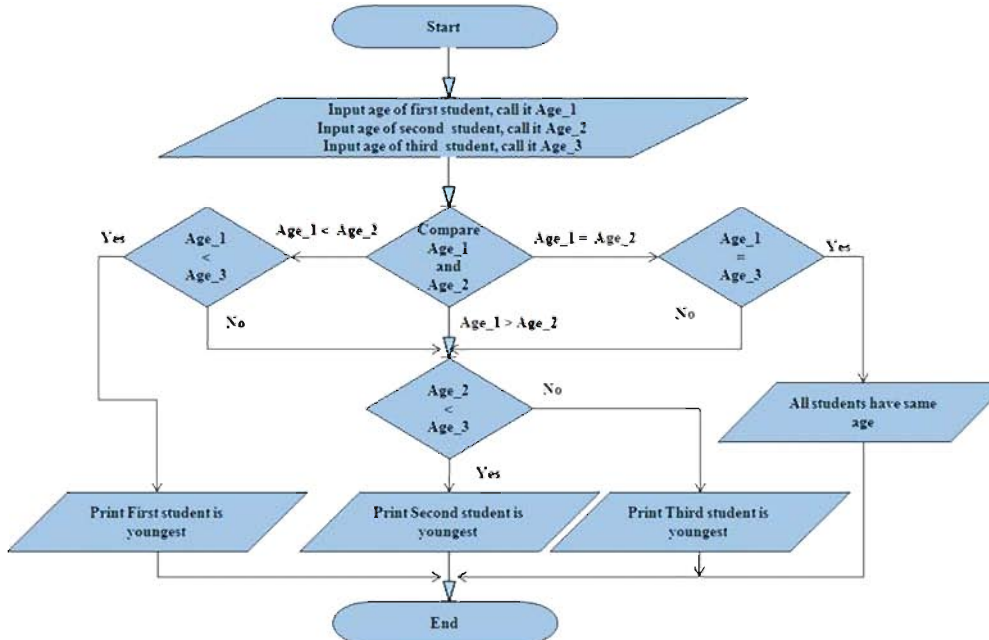
આકૃતિ 9.12 : બે વિદ્યાર્થીઓમાંથી નાનો વિદ્યાર્થી શોધવાનો ફ્લોચાર્ટ

ઉદાહરણ 5 : હવે આપણે ત્રણ વિદ્યાર્થીઓમાંથી સૌથી નાનો વિદ્યાર્થી શોધવાનો પ્રયત્ન કરીએ.

જવાબ :

સૌપ્રથમ ત્રણ વિદ્યાર્થીઓની ઉંમરનું ઇનપુટ લો અને તે ચલ Age_1, Age_2 અને Age_3માં રાખો. જો ત્રણે કિંમત એકસરખી હોય તો આપણી પાસે ત્રણે સરખી ઉંમરના વિદ્યાર્થી છે. આથી ત્રણે વિદ્યાર્થીઓને નાનામાં નાના ગણવામાં આવે. જો એમ ન હોય તો Age_1 અને Age_2ની સરખામણી કરો. જો Age_1ની કિંમત Age_2 કરતાં ઓછી હોય તો Age_1 ને Age_3 સાથે સરખાવો અને હજી Age_1 એ Age_3 કરતાં ઓછી હોય તો પહેલો વિદ્યાર્થી નાનામાં નાનો છે.

ઉપર જે સરખામણી કરી તેમાં જો Age_2 એ Age_1 કરતાં ઓછી હોય તો Age_2ને Age_3 સાથે સરખાવો અને જો Age_2 એ Age_3 કરતાં ઓછી હોય તો બીજો વિદ્યાર્થી નાનામાં નાનો છે. આ સિવાય ત્રીજો વિદ્યાર્થી નાનામાં નાનો છે. આ પ્રશ્નના ઉકેલનો ફ્લોચાર્ટ આકૃતિ 9.13માં આપેલો છે.



આકૃતિ 9.13 : ત્રણ વિદ્યાર્થીઓમાંથી નાનામાં નાનો વિદ્યાર્થી શોધવાનો ફ્લોચાર્ટ

ઉદાહરણ 6 : ચાલો, ઉપરના ઉદાહરણમાં હજી પણ સુધારો કરીએ અને વર્ગના બધા વિદ્યાર્થીઓમાંથી નાનામાં નાનો વિદ્યાર્થી શોધીએ. આ પ્રશ્નના ઉકેલ માટે નીચે જણાવેલાં પગલાંનો ઉપયોગ કરી શકાય :

પગલું 1 : પહેલો વિદ્યાર્થી લો અને તેની કોઈ પણ ઉંમર હોય, ધારો કે તે MINIMUM છે.

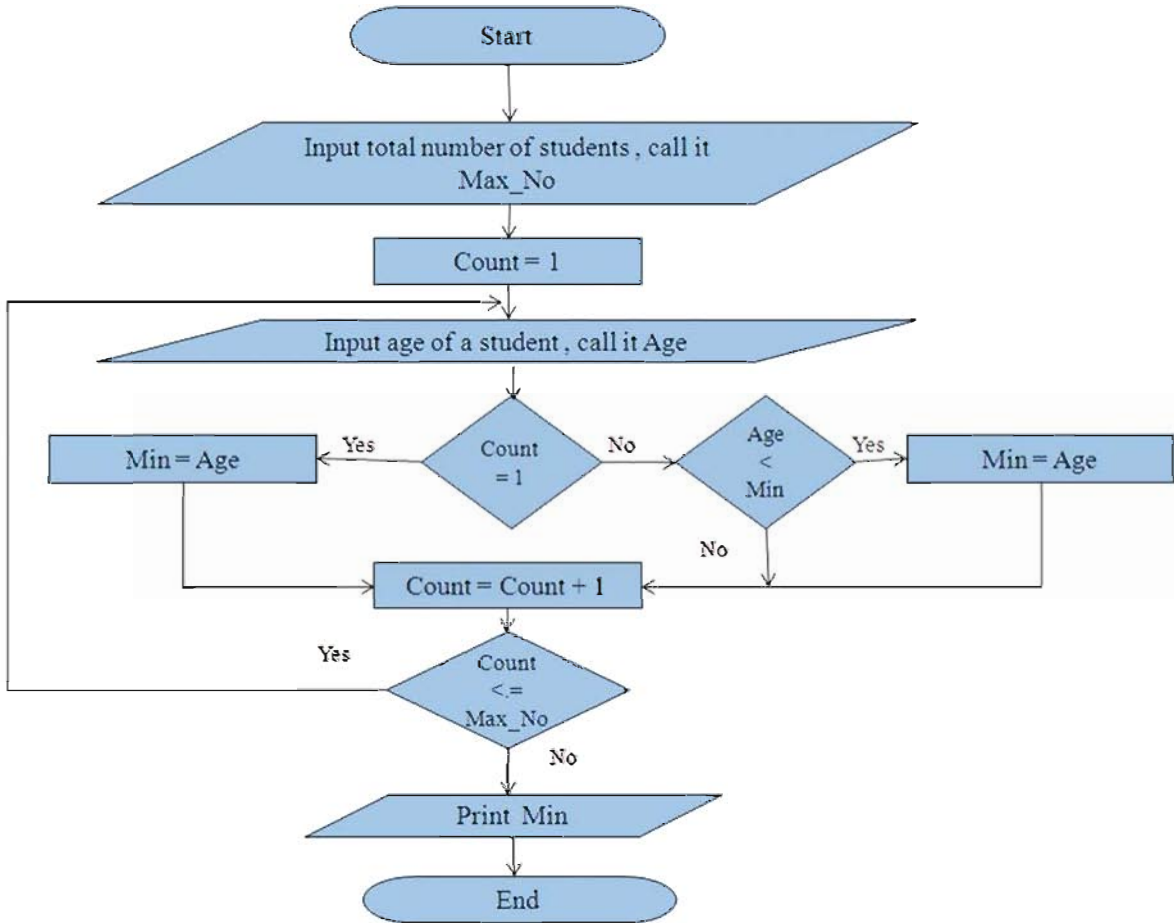
પગલું 2 : હવે બીજો વિદ્યાર્થી પસંદ કરો અને તેની ઉંમર MINIMUM જણાવેલ તેની સાથે સરખાવો.

પગલું 3 : જો નવા વિદ્યાર્થીની ઉંમર ઓછી હોય તો તેની ઉંમર MINIMUM બનાવીશું અને અગાઉના વિદ્યાર્થીની ઉંમર રદ કરીશું.

પગલું 4 : જો બંને વિદ્યાર્થીઓની ઉંમર એકસરખી હોય તો નવા વિદ્યાર્થીની ઉંમરની અવગણના કરવામાં આવશે અને અગાઉના વિદ્યાર્થીની ઉંમર હજી પણ MINIMUM છે.

પગલું 5 : આપણે ઉપર જણાવેલાં પગલાં 3 અને 4નું પુનરાવર્તન ચાલુ રાખીશું કે જ્યાં સુધી બધા વિદ્યાર્થીઓમાંથી નાનામાં નાનો વિદ્યાર્થી આપણને મળી જાય. જ્યારે અમુક વિદ્યાર્થીઓને એક કરતાં વધારે વખત પુનરાવર્તિત કરવામાં આવે તેને લૂપ (Loop) કહેવામાં આવે છે.

આ પ્રશ્નના ઉકેલનો ફ્લોચાર્ટ આકૃતિ 9.14માં આપેલો છે.



આકૃતિ 9.14 : વર્ગના બધા વિદ્યાર્થીઓમાંથી સૌથી યુવાન શોધવાનો ફ્લોચાર્ટ

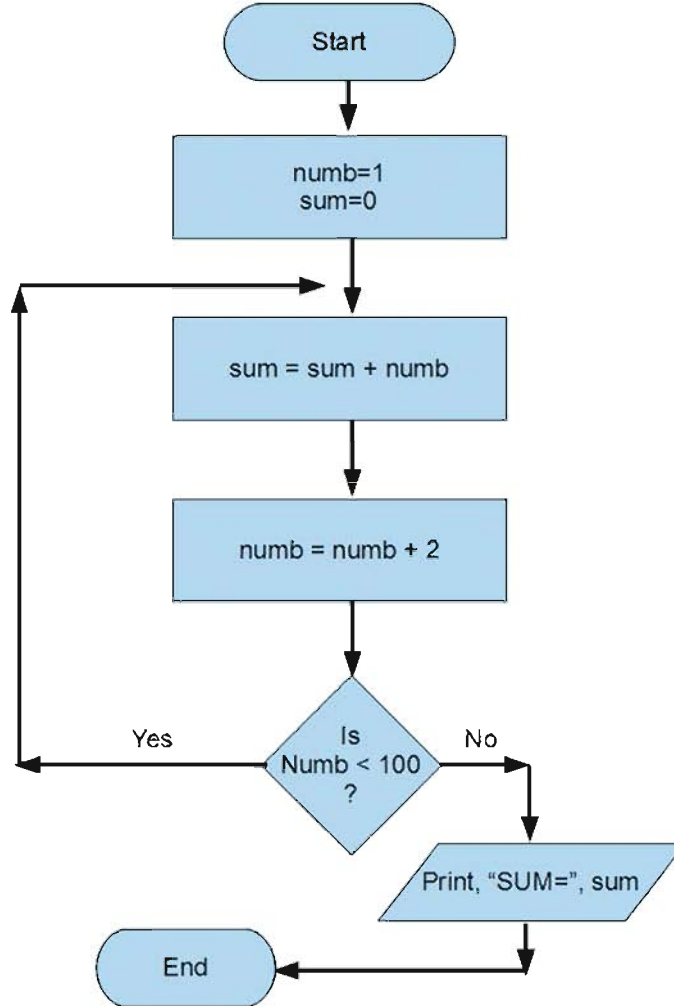
ઉદાહરણ 7 : ચાલો, હવે આપણે એક ગણિતનો પ્રશ્ન ઉકેલીએ. ધારો કે આપણે પહેલી 50 એકી સંખ્યા (odd numbers) ઓનો સરવાળો કરવા ઇચ્છીએ છીએ. એટલે કે આપણે $1 + 3 + 5 + 7 + \dots + 99$ નો સરવાળો કરવા ઇચ્છીએ છીએ.

જવાબ :

ધારો કે તમારી પાસે બે બોક્સ (પેટીઓ) છે, એક ડાબા હાથમાં અને બીજી જમણા હાથમાં. હવે ડાબી બાજુના બોક્સમાં એક લખોટી મૂકો અને હવે તે બોક્સને જમણા હાથમાં રહેલા બોક્સમાં ખાલી કરો. હવે ડાબા હાથમાં રહેલા બોક્સમાં ત્રણ લખોટી મૂકો અને તેને જમણા હાથમાં રહેલા બોક્સમાં ખાલી કરો (બદલી કરો). આથી જમણા હાથમાં રહેલા બોક્સમાં

કેટલી થશે ? જમણા હાથમાં રહેલા બોક્સમાં હવે કુલ $1 + 3$ લખોટીઓ હશે. જો આપણે આ પ્રક્રિયા $5, 7, \dots, 99$ લખોટીઓ સાથે ચાલુ રાખીશું તો જમણા હાથમાં રહેલા બોક્સમાં $1 + 3 + 5 + 7 + \dots + 99$ લખોટીઓ હશે. અહીં એ નોંધ કરો કે આપણે ધારેલું કે જમણા હાથનું બોક્સ શરૂઆતમાં ખાલી છે.

આ ક્રિયાને અનુરૂપ, આપણે બે ચલ `numb` અને `sum` લઈએ. `numb` ચલની શરૂઆતની કિંમત 1 અને `sum`ની કિંમત 0 (ખાલી બોક્સની જેમ) રાખીએ. હવે લૂપના દરેક સમયે `numb` ચલની કિંમત 2થી વધારવામાં આવે છે અને તેની કિંમત `sum`માં ઉમેરવામાં આવે છે. આથી `numb` ચલની કિંમત બદલાતી રહેશે અને તે 1, 3, 5, ... અને છેલ્લે 99 બનશે. આથી લૂપનો અંત આવે ત્યાં સુધી વિધાન `sum = sum + numb`થી `numb`ની કિંમત `sum`માં ઉમેરાતી રહેશે. આ પ્રશ્નનો ફ્લોચાર્ટ આકૃતિ 9.15માં આપેલો છે.



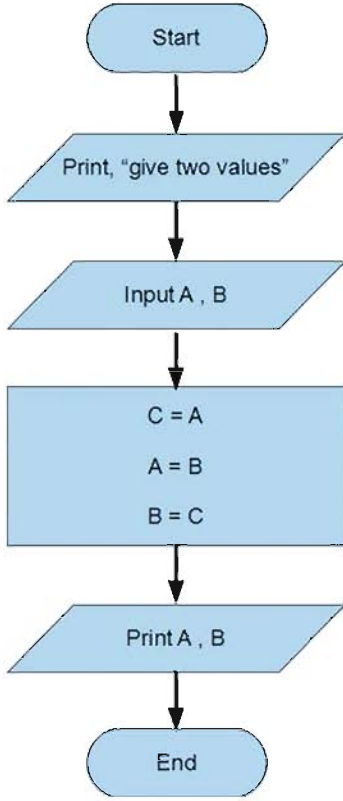
આકૃતિ 9.15 : 1થી 99 સુધીની એકી સંખ્યાઓના સરવાળાનો ફ્લોચાર્ટ

ઉદાહરણ 8 : હવે આપણે બે ચલની કિંમતની અદલાબદલી કરવાની બે જુદી જુદી રીતો જોઈએ.

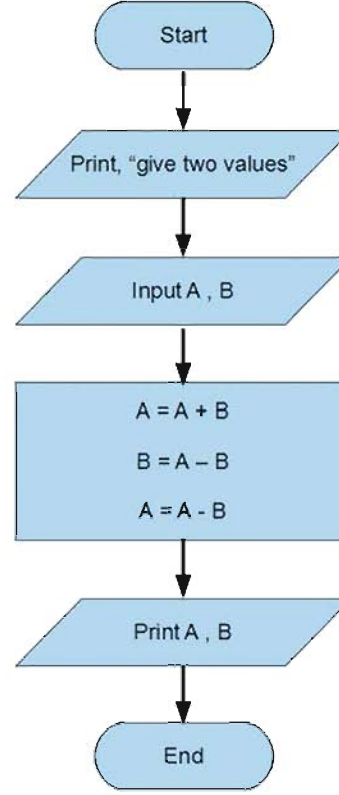
જવાબ :

બે ચલ `A` અને `B` લો. જો `A` અથવા `B`ને કોઈ અન્ય કિંમત વડે બદલવામાં આવશે તો તેમાં શરૂઆતમાં રહેલી કિંમત આપણે ગુમાવીશું. આથી આપણે એક વધારાનો ચલ `C`નો ઉપયોગ કરીશું. `A`ની કિંમત જળવાઈ રહે તે માટે તેની કિંમત પહેલાં આપણે `C`માં રાખીએ. તે પછી `B`ની કિંમત આપણે `A`માં લઈ જઈએ અને ત્યાર બાદ `A`ની સાચવેલી કિંમત (જે `C`માં રાખેલી છે) `B`માં લઈ જઈએ. આકૃતિ 9.16માં આ ઉકેલનો ફ્લોચાર્ટ આપેલો છે.

આકૃતિ 9.17માં આ જ પ્રશ્નના ઉકેલનો ફ્લોચાર્ટ આપેલો છે, પણ તેમાં વધારાના ચલનો ઉપયોગ થતો નથી. ખરેખર કિંમત બદલવા માટે સરવાળા અને બાદબાકીનો ઉપયોગ કરેલો છે.



**આકૃતિ 9.16 : વધારાના ચલનો ઉપયોગ કરીને
કિંમતની અદલાબદલી**



**આકૃતિ 9.17 : વધારાના ચલ
વિના કિંમતની અદલાબદલી**

ફ્લોચાર્ટ(Flowchart)ના ફાયદાઓ અને ગેરફાયદાઓ (Advantages and Disadvantages of Flowchart)

જ્યારે આપણે કોઈ પણ પ્રકારનાં ટૂલ્સ (tools) વાપરીએ છીએ, ત્યારે તેના કેટલાક ફાયદાઓ અને ગેરફાયદાઓ હોય છે. ફ્લોચાર્ટ તેમાં અપવાદ નથી. તેના કેટલાક ફાયદાઓ અને મર્યાદાઓ છે.

ફ્લોચાર્ટના ફાયદા :

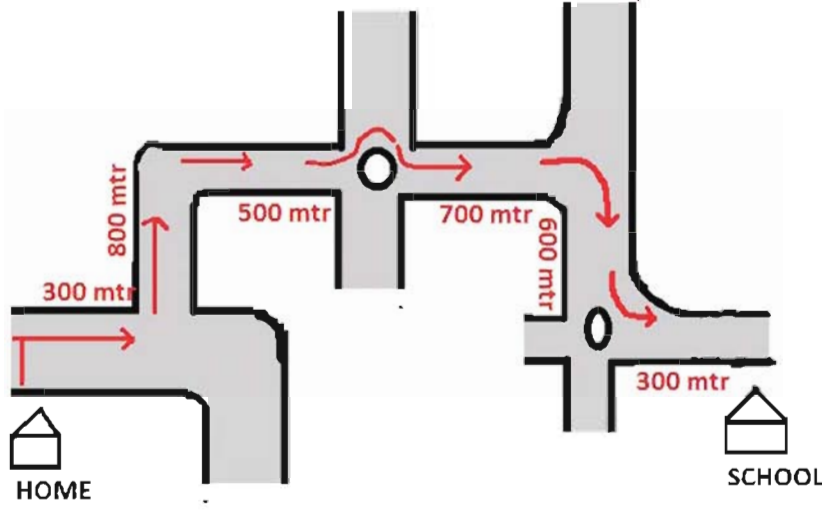
1. તે સમસ્યાનો ઉકેલ ચિત્રાત્મક સ્વરૂપે રજૂ કરતો હોવાથી સમજવો ખૂબ સરળ છે.
2. તે કમ્પ્યુટરની સૂચનાઓ લખવામાં ખૂબ જ અનુકૂળ મદદ પૂરી પાડે છે.
3. સમસ્યાના ઉકેલના પગલાંની સમીક્ષા કરવામાં અને સુધારવામાં પ્રોગ્રામરને સહાયભૂત બને છે.
4. સમસ્યાના ઉકેલની જુદી જુદી રીતો વિશે ચર્ચા કરવામાં તે મદદરૂપ બને છે, જેને કારણે અંતે તેની સરખામણી ભૂલરહિત કરવાની સગવડ તે પૂરી પાડે છે.

ફ્લોચાર્ટના ગેરફાયદા :

1. મોટી અને ગૂંચવાડાભારી સમસ્યાનો ફ્લોચાર્ટ દોરવાનું કામ ઘણો સમય માગી લે છે અને તે થકાવટવાળું કામ છે.
2. ફ્લોચાર્ટમાં અનેક ચિહ્નો હોય છે અને આથી પ્રોગ્રામના તર્કમાં કોઈ પણ ફેરફાર કરવાથી અનેક સમયે નવો ફ્લોચાર્ટ દોરવાની જરૂર પડે છે.
3. ફ્લોચાર્ટમાં ડેટા કેટલા વિગતવાર જણાવવા તેના કોઈ ધોરણ નથી. આથી એકસરખા તર્કવાળી સમસ્યાના ફ્લોચાર્ટ તેની અંદરની માહિતીથી અલગ પડે છે.

અલ્ગોરિથમ (Algorithm) (કલનવિધિ)

તમારો મિત્ર તમારા ઘેર આવી ગયો છે અને હવે તમારી નિશાળમાં આવવાની ઈચ્છા રાખે છે. તે તમને ફોન કરે છે અને તમારી નિશાળ સુધી પહોંચવાની દિશા પૂછે છે. સ્કૂલ કઈ રીતે પહોંચી શકાય તેની સૂચના આપવા માટે તમારા મગજમાં પહેલાં કોઈ નિયત માર્ગ હોય એ જરૂરી છે. ધારો કે, આકૃતિ 9.18માં દર્શાવ્યા પ્રમાણેનો નિયત માર્ગનો નકશો તેને મદદ કરવા માટે છે.



આકૃતિ 9.18 : નિયત માર્ગનો નકશો

જવાબ :

આકૃતિ 9.18માં આપેલાં નિયત માર્ગના નકશાનો ઉપયોગ કરીને તમે તમારા મિત્રને નીચે જણાવ્યા પ્રમાણે એક પછી એક પગલાંનું માર્ગદર્શન આપી શકશો :

પગલું 1 : મારા ઘરથી શરૂ કરી પહેલાં જમણી બાજુ ફરો અને 300 મીટર સીધા ચાલો.

પગલું 2 : હવે ડાબી બાજુ ફરો અને 800 મીટર સીધા જાઓ.

પગલું 3 : હવે જમણી બાજુ ફરો અને 500 મીટર ચાલો આથી ચાર રસ્તા પર એક સર્કલ આવશે.

પગલું 4 : સર્કલ પર વળાંક ન લો અને સીધા એ જ રસ્તા પર બીજા 700 મીટર ચાલો જેના પર તમે ચાલતા આવ્યા હતા.

પગલું 5 : હવે જમણી બાજુ ફરો અને 600 મીટર ચાલ્યા બાદ ફરી ચાર રસ્તા વચ્ચે અંડાકાર આકારના ફુવારા પર તમે પહોંચશો.

પગલું 6 : આ જંકશનથી ડાબી બાજુ ફરો અને 300 મીટર ચાલો.

પગલું 7 : હવે તમે શેરીની જમણી બાજુએ મારી નિશાળ જોઈ શકશો.

ઉપરનાં ઉદાહરણ પરથી આપણે કહી શકીએ કે પગલાંઓની શ્રેણીને અનુસરવાથી તમારી નિશાળે પહોંચવાનો પ્રશ્ન ઉકેલી શકીએ છીએ. કમ્પ્યુટર સાયન્સમાં અલ્ગોરિધમ કોઈ ચોક્કસ સમસ્યાના ઉકેલ માટેની ક્રમશઃ કાર્યપ્રણાલીનો નિર્દેશ કરે છે. અલ્ગોરિધમ અંગ્રેજી જેવી પ્રાકૃતિક ભાષામાં લખવામાં આવે છે. એક વાર અલ્ગોરિધમ લખવામાં આવે પછી તેના ઉપરથી કમ્પ્યુટર પ્રોગ્રામ લખવાનું અતિ સરળ બની જાય છે. ચાલો, હવે થોડાક અલ્ગોરિધમ લખવાનું શીખીએ.

ઉદાહરણ 1 :

1થી 100ની અંદર 11 વડે વિભાજિત સંખ્યાઓનો સરવાળો શોધવાનો અલ્ગોરિધમ લખો.

અલ્ગોરિધમ :

પગલું 1 : આરંભ

પગલું 2 : Num અને Sum બે ચલ લો.

પગલું 3 : શરૂઆતમાં કિંમત આપો. Num = 1 અને Sum = 0.

પગલું 4 : Num ચલ 11 વડે ભાગી શકાય તેમ છે કે કેમ તે તપાસો. આ માટે Numને 11 વડે ભાગીને શેષ શોધવી પડશે. આ શેષને અન્ય ચલ (ધારો કે) Rમાં સંગ્રહ કરો.

$$R = \text{Num} - \text{int}(\text{Num}/11) * 11$$

પગલું 5 : જો Rની કિંમત શૂન્ય ન હોય તો પગલું 7 પર જાઓ.

પગલું 6 : $Sum = Sum + Num$

પગલું 7 : $Num = Num + 1$

પગલું 8 : જો Num ની કિંમત 100 થી ઓછી હોય તો પગલું-4 પર જાઓ.

પગલું 9 : Sum પ્રિન્ટ કરો.

પગલું 10 : અંત

સમજૂતી :

શેષ શોધવા માટે આપણે Numમાં સંગ્રહ કરેલી સંખ્યાને 11 વડે ભાગીએ છીએ અને પછી તેનો પૂર્ણાંક ભાગ લઈએ છીએ. દા.ત. જો Numની કિંમત 39 હોય તો તેને 11 વડે ભાગવાથી આપણે 3.54545455 મેળવીએ. જો જવાબનો પૂર્ણાંક ભાગ લઈએ તો તે ફક્ત 3 છે. દશાંશ ચિહ્ન પછીનો ભાગ કાઢી નાખવામાં આવે છે. જો પૂર્ણાંક ભાગને 11 સાથે ગુણાકાર કરવામાં આવે તો આપણે 33 મેળવીશું. જો આ 33ને મૂળ સંખ્યા 39માંથી બાદ કરવામાં આવે તો આપણે કિંમત 6 મેળવીશું. આ કિંમત શેષ છે. અહીં એ નોંધ કરશો કે int નામની પ્રક્રિયા કે જે સંખ્યાનો પૂર્ણાંક ભાગ આપે છે, તેની આપણે ધારણા કરેલી છે.

પગલાં 1થી 4 જાતે સમજૂતી આપે તેવાં છે. પગલું 5 ઉપર જણાવેલી તકનીક વાપરીને શેષની કિંમત તપાસે છે. જો શેષ શૂન્ય ન હોય તો Numની કિંમત 11 વડે વિભાજિત નથી અને આપણે તે પછીની સંખ્યા લઈશું. જો Numની કિંમત 11 વડે વિભાજિત હોય તો આપણે પગલાં 6 ઉપર જઈશું. આપણે Numની કિંમત Sumમાં ઉમેરીશું અને પછી પગલાં 7નો અમલ કરીશું. પગલાં 8માં આપણે Num કિંમત 100 છે કે ઓછી તેની તપાસ કરીશું. આ તપાસ લૂપની બહાર નીકળવા માટે વપરાય છે. આપણે આ રીતે જ્યાં સુધી Num ની કિંમત 100 કરતાં ઓછી હશે ત્યાં સુધી ઉપરની ક્રિયા ચાલુ રાખીશું. અંતમાં આપણે Sum ની કિંમત છાપીશું.

ઉદાહરણ 2 : આપેલી મુદ્દલ રકમ, સમયગાળો અને વ્યાજ દર પરથી વાર્ષિક વ્યાજ ગણવામાં આવે ત્યારે ચક્રવૃદ્ધિ વ્યાજની ગણતરી કરવા માટેનો અલ્ગોરિધમ લખો.

અલ્ગોરિધમ :

આપણે ચક્રવૃદ્ધિ વ્યાજની ગણતરી માટે નીચેનું સૂત્ર વાપરીએ છીએ :

$$CI = P * (1 + R/100)^N - P$$

અહીં P-મુદ્દલ રકમ, R-વ્યાજનો દર અને N-વર્ષમાં સમયગાળો

પગલું 1 : આરંભ

પગલું 2 : ચાર ચલ P, N, R અને CI લો.

પગલું 3 : P, N અને Rની કિંમત આપો.

પગલું 4 : સૂત્ર $CI = P * (1 + R/100)^N - P$ વાપરી ચક્રવૃદ્ધિ વ્યાજ ગણો.

પગલું 5 : CIમાં સંગ્રહ કરેલી કિંમત છાપો.

પગલું 6 : અંત

ઉદાહરણ 3 :

કોઈ કર્મચારીના કામના કલાકોના આધારે અઠવાડિક પગાર શોધો. આ પ્રશ્નના જવાબ માટે આપણે કર્મચારીને એક કલાકનો કેટલો પગાર ચૂકવાય છે તે જાણવું પડે.

અલ્ગોરિધમ :

અઠવાડિક પગાર ચૂકવવા માટે આપણે બે કિંમતના ઇનપુટ (નિવેશ) લેવા પડે : કર્મચારીએ કેટલાં કલાક કામ કર્યું તે અને દરેક કલાકનો પગાર.

પગલું-1 : આરંભ

પગલું-2 : ત્રણ ચલ No_Of_Hrs_Worked, Pay_Per_Hour અને Weekly_Pay લો.

પગલું-3 : No_Of_Hrs_Worked અને Pay_Per_Hourની કિંમત આપો.

પગલું-4 : નીચેનું સૂત્ર વાપરી અઠવાડિક પગારની ગણતરી કરો :

$$\text{Weekly_Pay} = \text{No_Of_Hrs_Worked} * \text{Pay_Per_Hour}$$

પગલું-5 : ચલ Weekly_Payમાં સંગ્રહ કરેલી કિંમત છાપો.

પગલું-6 : અંત





સારાંશ

આ પ્રકરણમાં આપણે સમસ્યાના ઉકેલ માટે બે વધારે વપરાતી તકનિક ફ્લોચાર્ટ અને અલ્ગોરિધમ વિશે શીખ્યા. આપણે ફ્લોચાર્ટમાં વપરાતી વિવિધ સંજ્ઞાઓ વિશે પણ જાણ્યું. સમસ્યાના ઉકેલ માટે ફ્લોચાર્ટ વાપરવો કે અલ્ગોરિધમ તેના નિર્ણય માટે વપરાશકર્તાની જરૂરિયાતો બાબત જાણકારી મેળવી.

સ્વાધ્યાય

1. ફ્લોચાર્ટના મૂળભૂત અંગો કયા કયા છે ?
2. કયા પ્રકારની સમસ્યાનો ઉકેલ કમ્પ્યુટર વડે લાવી શકાય ?
3. કયા પ્રકારનાં કાર્યો એ કમ્પ્યુટરની પ્રક્રિયાના ભાગ છે ?
4. અંગગણિતનાં કાર્યોમાં કયા કયા કાર્યોનો સમાવેશ થાય છે ?
5. નિર્ણય પેટી(decision box)નો ઉપયોગ કયા પ્રકારનાં કાર્ય દર્શાવવા માટે થાય છે ?
6. ચલને વ્યાખ્યાયિત કરો.
7. અલ્ગોરિધમ એટલે શું ? તે ફ્લોચાર્ટથી કઈ રીતે જુદો પડે છે ?
8. આપેલા વિકલ્પોમાંથી ખરો વિકલ્પ પસંદ કરો :

(1) ફ્લોચાર્ટની શરૂઆત દર્શાવવા માટે કઈ સંજ્ઞાનો ઉપયોગ થાય છે ?

- (a) 
- (b) 
- (c) 
- (d) 

- (2) નીચેનામાંથી શું સમસ્યાના ઉકેલ માટે યોગ્ય ક્રમમાં સૂચનાઓનો નિર્દેશ કરે છે ?
 (a) અલગોરિધમ (b) ફ્લોચાર્ટ (c) શ્રેણી (d) રોડમેપ (નકશો)
- (3) નીચેનામાંથી કયું ચિહ્ન ફ્લોચાર્ટમાં શરત તપાસવા માટે વપરાય છે ?
 (a) હીરો (ડાયમંડ) (b) વર્તુળ (સર્કલ) (c) તીર (એરો) (d) ચોરસ (સ્કવેર)
- (4) નીચેનામાંથી કયું ચિહ્ન ફ્લોચાર્ટમાં આઉટપુટ દર્શાવવા માટે વપરાય છે ?
 (a) ચોરસ (સ્કવેર) (b) વર્તુળ (સર્કલ)
 (c) સમાંતર ચતુર્ભુજ (પેરેલેલોગ્રામ) (d) ત્રિકોણ (ટ્રાયએંગલ)
- (5) નીચેનામાંથી કયું ચિહ્ન ફ્લોચાર્ટમાં અંત દર્શાવવા માટે વપરાય છે ?
 (a) વર્તુળ (b) હીરો
 (c) ગોળ ખૂણાવાળો લંબચોરસ (રાઉન્ડેડ રેક્ટેન્ગલ) (d) ચોરસ
- (6) નીચેનામાંથી કયું વિધાન અલગોરિધમ અને ફ્લોચાર્ટનો હેતુ જણાવે છે ?
 (a) સ્મૃતિ(મેમરી)ની ક્ષમતા જાણવા
 (b) અંક પદ્ધતિનો આધાર (base) ઓળખવા
 (c) આઉટપુટ(નિર્ગમ)ને પ્રિન્ટર ઉપર મોકલવું
 (d) સમસ્યાને સંપૂર્ણપણે અને સ્પષ્ટપણે જણાવવી
- (7) નીચેનામાંથી કઈ સમસ્યાના ઉકેલની તકનિક નથી ?
 (a) સુડો કોડ (b) ફ્લોચાર્ટ (c) અલગોરિધમ (d) શ્રેણી
- (8) નીચેનામાંથી સમસ્યાના ઉકેલની ચિત્રાત્મક રજૂઆત કઈ છે ?
 (a) સુડો કોડ (b) ફ્લોચાર્ટ
 (c) અલગોરિધમ (d) કમ્પ્યુટર પ્રોગ્રામ
- (9) ફ્લોચાર્ટમાં તીરનું ચિહ્ન શું બતાવવા માટે વપરાય છે ?
 (a) કાર્યના પ્રવાહની દિશા (b) કાર્યનો ક્રમ
 (c) કાર્યની શરૂઆત (d) કાર્યની પૂર્ણાકૃતિ
- (10) નીચેનામાંથી શું કોઈ પણ કાર્યનો મુખ્ય ભાગ (core part) છે ?
 (a) ઇનપુટ (નિવેશ) (b) આઉટપુટ (નિર્ગમ) (c) પ્રક્રિયા (d) અલગોરિધમ
- (11) નીચેનામાંથી કયું ચિહ્ન પ્રક્રિયા દર્શાવે છે ?
 (a) લંબચોરસ (b) ચોરસ (c) વર્તુળ (d) હીરો
- (12) નીચેનામાંથી ફ્લોચાર્ટમાં જુદા જુદા અનુસંધાન ચિહ્નની જોડી માટે શું વપરાય છે ?
 (a) તીર (b) મૂળાક્ષરો કે બીજા અક્ષર
 (c) વર્તુળ (d) હીરાનું ચિહ્ન

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો માટે ફ્લોચાર્ટ દોરો અને અલ્ગોરિથમ લખો :

1. આપેલા મીટરને સેન્ટીમીટરમાં ફેરવો.
2. આપેલા સેન્ટિગ્રેડને ફેરનહીટમાં ફેરવો.
3. આપેલી બે સંખ્યામાંથી નાની સંખ્યા શોધો.
4. આપેલી બે સંખ્યામાંથી નાની સંખ્યા શોધો. એ એકી સંખ્યા છે કે બેકી સંખ્યા તે તપાસો.
5. આપેલી ત્રણ સંખ્યામાંથી લઘુત્તમ સંખ્યા શોધો.
6. આપેલી કોઈ સંખ્યા અન્ય કોઈ આપેલી સંખ્યાથી વિભાજિત છે કે કેમ તે તપાસો.
7. ત્રણ સંખ્યાની સરેરાશ શોધો.
8. કોઈ વિદ્યાર્થીના કુલ ગુણને આધારે શોધો કે તે વિદ્યાર્થી પાસ છે કે નાપાસ (જો કુલ ગુણ 35થી ઓછા હોય તો વિદ્યાર્થીને નાપાસ જાહેર કરો).
9. કોઈ વિદ્યાર્થીના કુલ ગુણને આધારે વિદ્યાર્થીએ કઈ શ્રેણી મેળવી છે તે શોધો. (જો વિદ્યાર્થીના કુલ ગુણ 35થી ઓછા હોય તો વિદ્યાર્થીને D શ્રેણી મળે છે. જો કુલ ગુણ 35 અને 60 વચ્ચે હોય તો C શ્રેણી અને જો કુલ ગુણ 60 અને 70 વચ્ચે હોય તો B શ્રેણી અને 70 કરતાં વધારે ગુણ હોય તો A શ્રેણી મેળવે છે.)
10. X ₹ મુદ્દલ રકમની લોન, R% વ્યાજનો દર અને N વર્ષના સમયગાળાનાં ચક્રવૃદ્ધિ વ્યાજની ગણતરી કરો.
11. આપેલી સંખ્યાના અંકોનો સરવાળો કરો. ઉદાહરણ તરીકે, જો આપેલી સંખ્યા 8327 હોય તો આઉટપુટ $(8 + 3 + 2 + 7) = 20$ થશે.





સી ભાષાનો પરિચય

નવમા પ્રકરણમાં આપણે ફ્લોચાર્ટ અને અલ્ગોરિધમ વિશે અભ્યાસ કર્યો. આપણે અલ્ગોરિધમની રચના પણ શીખ્યા. ફ્લોચાર્ટ અને અલ્ગોરિધમ એ કોઈ પણ પ્રોગ્રામના ઉકેલ માટેનાં પાયારૂપ પગલાં છે. ત્યાર બાદ આ ફ્લોચાર્ટ કે અલ્ગોરિધમને પ્રોગ્રામમાં ફેરવવામાં આવે છે. પ્રોગ્રામ લખવા માટે અનેક ભાષાઓ ઉપલબ્ધ છે. ‘સી’ આવી જ એક ભાષા છે. આ પ્રકરણમાં આપણે ‘સી’ ભાષા વિશે પરિચય મેળવીશું.

પ્રોગ્રામિંગ ભાષાની જરૂરિયાત (Need of programming language)

આપણને સૌપ્રથમ પ્રશ્ન એ થાય કે, પ્રોગ્રામિંગ ભાષાની જરૂરિયાત શા માટે છે? પહેલેથી આપણે જેને જાણીએ છીએ તેવી અંગ્રેજી કે ગુજરાતી ભાષાનો ઉપયોગ કરીને પ્રોગ્રામ શા માટે લખી ન શકાય? જવાબ એ છે કે, આ ભાષાઓમાં વાક્યોના એક જ અર્થ હોતા નથી. આથી તે કમ્પ્યુટર સાથે સુનિશ્ચિત રીતે કાર્ય કરી શકે નહીં. કમ્પ્યુટર દ્વારા અપેક્ષિત કાર્ય કરાવવા માટે દરેક વાક્ય અર્થપૂર્ણ અને સુનિશ્ચિત હોવું જરૂરી છે.

પ્રોગ્રામિંગ ભાષાનો ઉપયોગ કરી લખવામાં આવેલી સૂચનાઓનો માત્ર એક જ અર્થ હોય છે. તે પૂર્વનિર્ધારિત નિયમોનો ગણ ધરાવે છે. આ નિયમો પ્રોગ્રામિંગ ભાષા માટે વાક્યરચના(syntax)ની રચના કરે છે. આમ, પ્રોગ્રામિંગ ભાષા શીખવી એટલે એવી નવી વાક્યરચના શીખવી, જેના દ્વારા કમ્પ્યુટરને આપવામાં આવનાર સૂચનાઓ નિશ્ચિતપણે રજૂ કરી શકાય. આ કાર્ય કોઈ પણ નવી ભાષા શીખતી વખતે તેના વ્યાકરણને શીખવા સમાન છે.

અનુવાદકની જરૂરિયાત (Need of translator)

આપણા દ્વારા બોલાયેલી ભાષાને કે પ્રોગ્રામિંગ ભાષાને પણ કમ્પ્યુટર સમજી શકતું નથી. તે માત્ર 0 અને 1ની ભાષા સમજે છે. આવી મુશ્કેલીનો સામનો આપણે ક્યારેક આપણા જીવનમાં પણ કરતા હોઈએ છીએ. માનો કે X અને Y બે વ્યક્તિઓ છે, જે એકબીજાની ભાષા જાણતી નથી. વ્યક્તિ-X ગુજરાતી ભાષા જાણે છે જ્યારે વ્યક્તિ-Y હિન્દી ભાષા જાણે છે. આ બે વ્યક્તિઓ કેવી રીતે એકબીજા સાથે વાતચિત કરશે? આના વિકલ્પ તરીકે એક ત્રીજી વ્યક્તિ-Z જરૂરી છે, જે આ બંને ભાષાની જાણકાર હોય. હવે, વ્યક્તિ-Xને વ્યક્તિ-Y સુધી કોઈ સંદેશ પહોંચાડવો હશે ત્યારે, તે પ્રથમ ગુજરાતી ભાષામાં વ્યક્તિ-Zને સંદેશો કહેશે. વ્યક્તિ-Z તે સંદેશનો હિન્દીમાં અનુવાદ કરશે અને તે સંદેશો વ્યક્તિ-Yને પહોંચાડશે. વ્યક્તિ-Y જ્યારે વ્યક્તિ-X સાથે વાતચિત કરવા ઇચ્છે ત્યારે આ જ પ્રક્રિયાનો વિપરીત ક્રમમાં અમલ કરવામાં આવશે. અહીં વ્યક્તિ-Zને અનુવાદક (translator) તરીકે ઓળખવામાં આવે છે. તથા એક ભાષાને અન્ય ભાષામાં ફેરવવાની ક્રિયાને અનુવાદ (translation) કહેવામાં આવે છે.

આપણી ભાષા નહીં સમજી શકવાની કમ્પ્યુટરની સમસ્યાનું નિરાકરણ અનુવાદક (translator) પ્રકારના સૉફ્ટવેર પ્રોગ્રામ દ્વારા મેળવવામાં આવે છે. આ અનુવાદકને કંપાઈલર (compiler) તરીકે ઓળખવામાં આવે છે. અનુવાદની પ્રક્રિયા આ પ્રકરણના અંતમાં સમજાવવામાં આવેલી છે.

પ્રોગ્રામ અને પ્રોગ્રામની લાક્ષણિકતાઓ (Program and characteristics of program)

કમ્પ્યુટર પાસેથી કોઈ પૂર્વ વ્યાખ્યાયિત કાર્ય કરાવવા માટે આપવી પડતી ચોક્કસ અને નિશ્ચિત સૂચનાઓના સમૂહને પ્રોગ્રામ કહેવામાં આવે છે. પસંદ કરેલ ભાષા દ્વારા આ સૂચનાઓને ક્રમબદ્ધ રીતે લખવાની પ્રક્રિયાને પ્રોગ્રામિંગ તરીકે ઓળખવામાં આવે છે.

એક સારો પ્રોગ્રામ નીચે જણાવેલ લાક્ષણિકતાઓ ધરાવતો હોવો જોઈએ :

1. નિશ્ચિત પગલાં પછી પ્રોગ્રામનો અંત આવે તે જરૂરી છે.
2. પ્રોગ્રામમાં આવેલ સૂચનાઓ ચોક્કસપણે વ્યાખ્યાયિત થવી જોઈએ, એટલે કે તે એકથી વધુ અર્થ ધરાવતી ન હોવી જોઈએ.
3. તમામ સૂચનાઓ અસરકારક હોવી જોઈએ, એટલે કે તેનો અમલ યોગ્ય રીતે થવો જોઈએ.

4. પ્રોગ્રામ શૂન્ય કે વધુ ઇનપુટ લઈ શકે.
5. પ્રોગ્રામ એક કે વધુ આઉટપુટ આપી શકે.

તમે જોઈ શકો છો કે, આ લાક્ષણિકતાઓ અલ્ગોરિથમની લાક્ષણિકતાઓને મળતી આવે છે. ઉદાહરણ 10.1 એક નમૂનારૂપ સી પ્રોગ્રામ દર્શાવે છે. ઉદાહરણ 10.1 માટેનું કોડ લિસ્ટિંગ અને પરિણામ આકૃતિ 10.1માં દર્શાવ્યા છે. SciTE એડિટરના ઉપયોગની કાર્ય પદ્ધતિ આપણે આ પ્રકરણના અંતમાં શીખીશું.



```

1 10_1.c
/* Example 1: My first C program */
#include <stdio.h>
int main( )
- {
    printf("Welcome to the world of C programming using Scite \n");
    return 0;
}

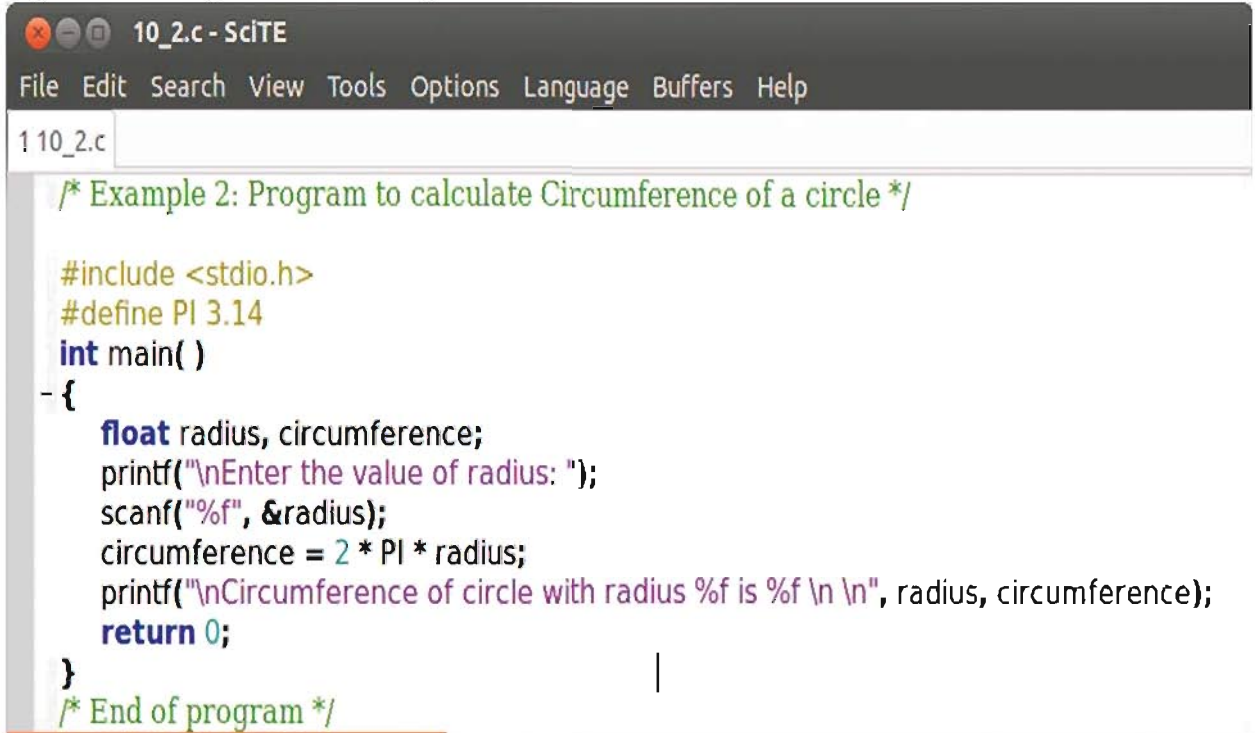
>gcc -pedantic -Os -std=c99 10_1.c -o 10_1
>Exit code: 0
>./10_1
Welcome to the world of C programming using Scite
>Exit code: 0

```

આકૃતિ 10.1 : પ્રથમ ‘સી’ પ્રોગ્રામ

આ પ્રોગ્રામ દ્વારા "Welcome to the world of C programming using Scite" સંદેશ દર્શાવવામાં આવશે. અહીં main()ને ઉપયોગકર્તા દ્વારા નિર્મિત (user defined) વિધેય કહેવામાં આવે છે જ્યારે printf() એ આંતર પ્રસ્થાપિત (inbuilt) કે લાઈબ્રેરી વિધેય છે. ઉપયોગકર્તા દ્વારા તેની જરૂરિયાત અનુસાર રચવામાં આવેલા વિધેયને ઉપયોગકર્તા નિર્મિત વિધેય (User Defined Function) કહે છે. ઉપયોગકર્તા પોતાના પ્રોગ્રામમાં ઉપયોગ કરી શકે તેવા ‘સી’ ભાષામાં પહેલેથી ઉપલબ્ધ વિધેયને આંતરસ્થાપિત કે લાઈબ્રેરી વિધેય તરીકે ઓળખવામાં આવે છે.

સી ભાષાના વિવિધ ઘટકોની વિસ્તૃત ચર્ચા શરૂ કરતાં પહેલાં આકૃતિ 10.2માં દર્શાવેલ પ્રોગ્રામનો અભ્યાસ કરીએ જેનું પરિણામ આકૃતિ 10.3માં દર્શાવ્યું છે.



```

1 10_2.c
/* Example 2: Program to calculate Circumference of a circle */
#include <stdio.h>
#define PI 3.14
int main( )
- {
    float radius, circumference;
    printf("\nEnter the value of radius: ");
    scanf("%f", &radius);
    circumference = 2 * PI * radius;
    printf("\nCircumference of circle with radius %f is %f \n \n", radius, circumference);
    return 0;
}
/* End of program */

```

આકૃતિ 10.2 ઉદાહરણ 10.2નું કોડ-લિસ્ટિંગ


```
harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ ./a.out
Enter the value of radius: 2.5

Circumference of circle with radius 2.500000 is 15.700000
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$
```

આકૃતિ 10.3 : ઉદાહરણ 10.2નું પરિણામ

સમજૂતી

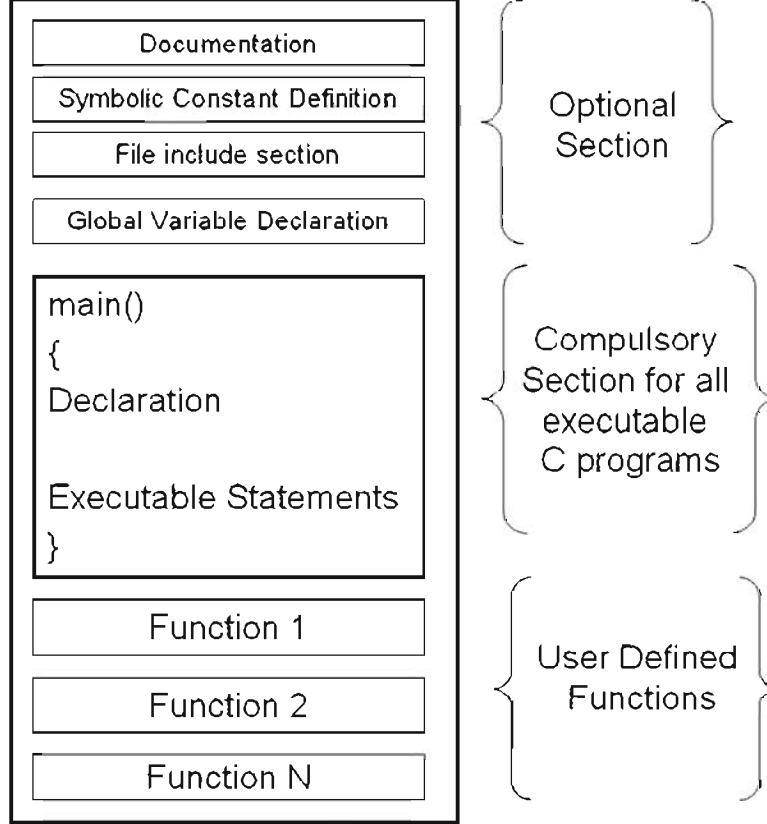
અહીં આપેલ પ્રોગ્રામનો ઉપયોગ કરી આપેલ ત્રિજ્યાને આધારે વર્તુળનો પરિઘ શોધી શકાય છે.

- પ્રથમ વિધાન એવી નોંધ (comment) છે, જે પ્રોગ્રામના ઉદ્દેશ્યનો નિર્દેશ કરે છે.
- બીજા વિધાન દ્વારા કંપાઈલરને "stdio.h" નામની હેડર ફાઈલની વિગતોનો સમાવેશ કરવાની સૂચના આપવામાં આવી છે. પ્રોગ્રામમાં ઉપયોગમાં લેવામાં આવેલા આંતર પ્રસ્થાપિત વિધેય વિશેની માહિતી આ ફાઈલમાં આપેલી હોય છે.
- ત્રીજું વિધાન PI નામના સાંકેતિક અચળ (Symbolic Constant)ને વ્યાખ્યાયિત કરે છે, જેની કિંમત 3.14 છે. આ વિધાનને પ્રી-પ્રોસેસર ડાઈરેક્ટિવ (pre-processor directive) તરીકે ઓળખવામાં આવે છે. પ્રોગ્રામનો અમલ કરતાં પહેલાં આ વિધાન દ્વારા તમામ PI શબ્દને 3.14 કિંમત સાથે બદલવાની સૂચના આપવામાં આવે છે.
- ચોથું વિધાન એ ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત વિધેય main() છે, જે અમલ થઈ શકે તેવા (executable) તમામ પ્રોગ્રામ માટે ફરજિયાત છે. અહીં આ વિધેય પૂર્ણાંક સંખ્યા પરત કરતું હોવાથી int main() સ્વરૂપે લખવામાં આવ્યું છે.
- પાંચમું વિધાન main() વિધેયની શરૂઆત છે.
- છઠ્ઠું વિધાન radius અને circumference નામના બે ચલને વ્યાખ્યાયિત કરે છે. આ ચલ અપૂર્ણાંક સંખ્યાઓનો સંગ્રહ કરવા સક્ષમ છે. સી ભાષા જુદી જુદી કિંમતોનો સંગ્રહ કરવા માટે સક્ષમ છે. સી ભાષામાં જુદી જુદી કિંમતોનો સંગ્રહ કરવા માટે સક્ષમ એવા ઘટકોને ચલ (variable) તરીકે ઓળખવામાં આવે છે.
- સાતમું વિધાન સંદેશ છાપવા માટે printf વિધાનનો ઉપયોગ કરે છે.
- આઠમા વિધાનમાં આંતર પ્રસ્થાપિત વિધેય scanf દ્વારા ઉપયોગકર્તા પાસેથી ત્રિજ્યા (radius) મેળવવામાં આવે છે.
- નવમું વિધાન એક સી એક્સપ્રેશન (C expression) છે, તે વર્તુળનો પરિઘ ગણવા માટેનું સમીકરણ છે.
- દસમું વિધાન ત્રિજ્યા (radius) અને પરિઘ(circumference)ની કિંમતો સાથે સ્ક્રીન પર સંદેશ દર્શાવે છે.
- અગિયારમું વિધાન વિધેયમાંથી સામાન્ય બર્હિગમન (exit) પૂરું પાડે છે. જો આ વિધાન લખવામાં ન આવે તો ચેતવણીનો સંદેશ - "Function should return a value" દર્શાવવામાં આવે છે. પ્રોગ્રામના કાર્યમાં તે કોઈ અસર કરતું નથી, પરંતુ સામાન્ય બર્હિગમન એ હંમેશા એક સારો વિકલ્પ છે.
- બારમું વિધાન main() વિધેયનો અંત દર્શાવે છે.

આકૃતિ 10.1 અને 10.2માં દર્શાવ્યા મુજબ main() વિધેય સી પ્રોગ્રામના સંપૂર્ણ ભાગનું બંધારણ કરે છે. વધુમાં આપણે એમ કહી શકીએ કે અમલ-યોગ્ય તમામ સી પ્રોગ્રામમાં ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત એવું એક main() નામનું વિધેય હોવું જરૂરી છે. દરેક સી પ્રોગ્રામનું અમલીકરણ main() વિધેયના ઉઘડતા છગડિયા કૌંસ ({)થી શરૂ કરવામાં આવે છે.

સી પ્રોગ્રામનું માળખું (Structure of C program)

અગાઉ આપણે જોઈ ગયા તે મુજબ સી પ્રોગ્રામ એ વિધેય નામે ઓળખાતા સમૂહોનો ગણ છે. વિધેય એ એક કે વધુ વિધાનોનો સમૂહ છે, જેના દ્વારા પૂર્વનિર્ધારિત કાર્યોનો અમલ કરી શકાય છે. આકૃતિ 10.4માં સંપૂર્ણ સી પ્રોગ્રામનું માળખું દર્શાવ્યું છે. આ વિભાગોની વિસ્તૃત ચર્ચા કરીએ.



આકૃતિ 10.4 : સંપૂર્ણ સી પ્રોગ્રામનું માળખું

દસ્તાવેજીકરણ વિભાગ (Documentation section)

આ એક મરજિયાત વિભાગ છે. નામ દર્શાવે છે તે મુજબ આ વિભાગનો ઉપયોગ દસ્તાવેજીકરણ માટે કરવામાં આવે છે. તે /* અને */ ની વચ્ચે આવરીને લખવામાં આવે છે. કોઈ પણ સ્થાને /* અને */ ની વચ્ચે આવરીને લખવામાં આવેલા લખાણને ‘નોંધ’ (comment) માનવામાં આવે છે. કોમેન્ટ પર સી કંપાઈલર દ્વારા કોઈ પ્રક્રિયા કરવામાં આવતી નથી અને તેને જેમની તેમ છોડી દેવામાં આવે છે.

પ્રોગ્રામનો હેતુ, લેખકનું નામ, પ્રોગ્રામ લખ્યાની તારીખ વગેરે જેવી માહિતી દર્શાવવા માટે સામાન્ય રીતે આ વિભાગનો ઉપયોગ કરવામાં આવે છે. સી પ્રોગ્રામમાં કોઈ પણ સ્થાને કોમેન્ટ ઉમેરી શકાય છે. કોમેન્ટ ઉમેરવી એ હંમેશાં એક સારી ટેવ ગણાય છે, કારણકે તેના દ્વારા પ્રોગ્રામની વાંચનક્ષમતા અને સમજણ વધે છે.

સાંકેતિક અચળની વ્યાખ્યા (Symbolic constant definition)

આકૃતિ 10.2ના ઉદાહરણમાં દર્શાવ્યા મુજબ પ્રોગ્રામમાં PI નામના સાંકેતિક અચળનો ઉપયોગ કરવામાં આવ્યો છે. પ્રોગ્રામમાં આ પ્રકારના સાંકેતિક અચળનો ઉપયોગ કરવા માટે અચળની આગળ #define ઉમેરવું જરૂરી છે. અહીં, #defineને પ્રી-પ્રોસેસર ડાઈરેક્ટિવ (pre-processor directive) તરીકે ઓળખવામાં આવે છે. તે પ્રોગ્રામમાં આવતા તમામ સાંકેતિક અચળોને આપવામાં આવેલી કિંમત સાથે બદલવાની સૂચના કંપાઈલરને પૂરી પાડે છે. સામાન્ય રીતે સાંકેતિક અચળોને કેપિટલ અક્ષરો દ્વારા વ્યાખ્યાયિત કરવામાં આવે છે. આમ કરવાથી તેને સામાન્ય ચલ કરતાં અલગ પાડી શકાય છે.

ફાઈલ ઉમેરણ વિભાગ (File include section)

સી ભાષા આંતરપ્રસ્થાપિત કે લાઈબ્રેરી વિધેયો પૂરા પાડે છે. pow(), sqrt() વગેરે તેનાં ઉદાહરણ છે. આ વિધેયોને પૂર્વનિશ્ચિત ઉદ્દેશો હોય છે, જેમ કે, pow() નો ઉપયોગ x ની આપેલ ઘાત જેટલી કિંમત શોધવા માટે થાય છે તથા આપેલ ઘાત જેટલી કિંમત

શોધવા માટે થાય છે તથા આપેલ સંખ્યાનું વર્ગમૂળ શોધવા માટે `sqrt()` વિધેયનો ઉપયોગ કરવામાં આવે છે. જો જરૂર જણાય તો આપણે પ્રોગ્રામમાં આ વિધેયનો ઉપયોગ કરી શકીએ છીએ. આ વિધેયનો ઉપયોગ કરવા માટે તેની માહિતી ધરાવતી ફાઈલોને પ્રોગ્રામમાં ઉમેરવી પડે છે. સી ભાષામાં આ બધી ફાઈલોને હેડર ફાઈલ (header file) તરીકે ઓળખવામાં આવે છે. હેડર ફાઈલનું અનુલંબન ".h" હોય છે. પ્રોગ્રામમાં હેડર ફાઈલ ઉમેરવા માટે `#include <filename.h>` વાક્ય રચનાનો ઉપયોગ કરવામાં આવે છે. સી ભાષામાં ઉપલબ્ધ અનેક હેડર ફાઈલની યાદી તથા તેના ઉપયોગ અંગેની માહિતી પરિશિષ્ટ IV માં આપવામાં આવી છે.

વૈશ્વિક ચલ ઘોષણા વિભાગ (Global variable declaration section)

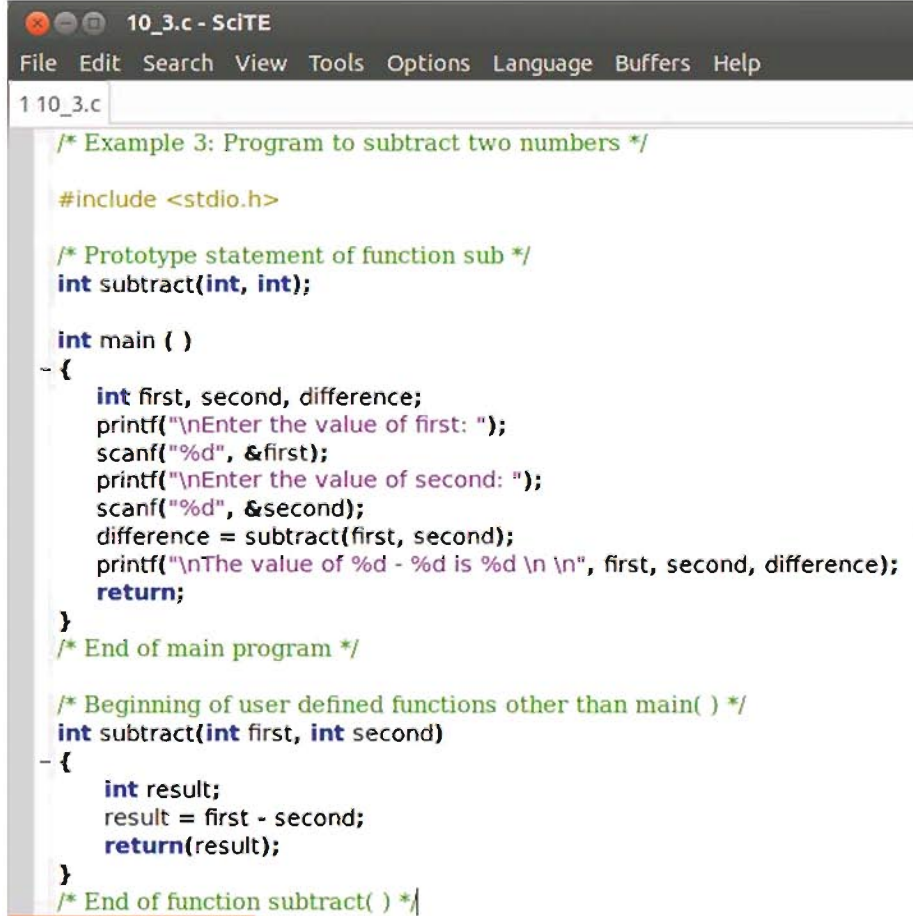
સી ભાષામાં ચલનો અમલ તેના વિસ્તાર (scope) પ્રમાણે કરવામાં આવે છે. ઉદાહરણ તરીકે બંધ થતા છગડિયા કૌંસ ({}) દ્વારા સી ચલનો વિસ્તાર નક્કી કરવામાં આવે છે. છગડિયા કૌંસમાં વ્યાખ્યાયિત કરેલા ચલને સ્થાનિક ચલ (local variable) તરીકે ઓળખવામાં આવે છે. જેમ કે, ઉદાહરણ 10.2માં ઉપયોગમાં લેવામાં આવેલા `radius` અને `circumference` ચલ `main()` વિધેયના સ્થાનિક ચલ છે. આ પ્રકારના ચલનો ઉપયોગ તેના વિસ્તારની બહાર કરી શકાતો નથી. જો આપણે ચલનો ઉપયોગ તેના વિસ્તારની બહાર પણ તમામ વિધેયોમાં કરવા ઇચ્છતા હોઈએ તો, તે પ્રકારના ચલને વૈશ્વિક ચલ (global variable) કહે છે. આ પ્રકારના ચલને તમામ વિધેયની પહેલા વ્યાખ્યાયિત કરવામાં આવે છે.

main વિધેય (main function)

તમામ સી પ્રોગ્રામમાં `main()` નામનું વિધેય આવેલું હોય છે. આ વિધેયથી સી પ્રોગ્રામનું અમલીકરણ શરૂ કરવામાં આવે છે. પ્રોગ્રામનું નિયંત્રણ સૌપ્રથમ આ વિધેયને મોકલવામાં આવે છે અને ત્યાંથી અન્ય ક્રિયાઓનો અમલ કરવામાં આવે છે.

ઉપયોગકર્તા નિર્મિત વિધેય (User defined functions)

સી ભાષા કોઈ પણ પ્રોગ્રામને નાના-નાના વિભાગોમાં વહેંચવાની સુવિધા પૂરી પાડે છે. આ વિભાગોને વિધેય કહે છે. આપણા પ્રોગ્રામમાં જરૂરી એવા તમામ અતિરિક્ત વિધેયને આ વિભાગમાં વ્યાખ્યાયિત કરવામાં આવે છે. આકૃતિ 10.5માં દર્શાવેલા ઉદાહરણ 10.3માં ઉપયોગકર્તા દ્વારા નિર્મિત એવા બે વિધેયોનો ઉપયોગ કરવામાં આવ્યો છે.



```

10_3.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 10_3.c

/* Example 3: Program to subtract two numbers */

#include <stdio.h>

/* Prototype statement of function sub */
int subtract(int, int);

int main ( )
- {
    int first, second, difference;
    printf("\nEnter the value of first: ");
    scanf("%d", &first);
    printf("\nEnter the value of second: ");
    scanf("%d", &second);
    difference = subtract(first, second);
    printf("\nThe value of %d - %d is %d\n\n", first, second, difference);
    return;
}
/* End of main program */

/* Beginning of user defined functions other than main( ) */
int subtract(int first, int second)
- {
    int result;
    result = first - second;
    return(result);
}
/* End of function subtract( ) */

```

આકૃતિ 10.5 : ઉદાહરણ 10.3નું કોડ લિસ્ટિંગ

સમજૂતી

આ પ્રોગ્રામ આપેલ બે સંખ્યાઓની બાદબાકી કરવા માટે ઉપયોગી છે. નાનો પ્રોગ્રામ હોવા છતાં તેને ઉપયોગકર્તા દ્વારા નિર્મિત બે વિધેયમાં વિભાજિત કરવામાં આવ્યો છે. પ્રથમ વિધેય `main()` બાદબાકી કરવા માટેની બે કિંમતો સ્વીકારે છે અને બીજું વિધેય `subtract()` એ બાદબાકીની ક્રિયા ખરેખર પાર પડે છે.

અહીં એ જોઈ શકાય છે કે `subtract()` વિધેયનો ઉપયોગ કરતાં પહેલાં તેની પ્રતિકૃતિ (prototype) રજૂ કરવામાં આવી છે. જો ઉપયોગકર્તા દ્વારા નિર્મિત વિધેયને `main()` વિધેયની નીચે લખવામાં આવ્યું હોય તો સી ભાષામાં આ જરૂરી છે. `main()` વિધેય ત્રણ પૂર્ણાંક ચલ ઘોષિત કરે છે અને તેમાંથી બે ચલની કિંમત ઉપયોગકર્તાને પૂછે છે. ત્યાર પછી `difference = subtract(first, second)` વિધાનનો ઉપયોગ કરી આ કિંમતોને `subtract` વિધેયને મોકલવામાં આવે છે. આ `first` અને `second` ને `subtract()` વિધેયના પ્રાયલ (parameters) કહે છે. આ વિધાનના અમલથી પ્રોગ્રામનું નિયંત્રણ `subtract()` વિધેયને મોકલવામાં આવે છે. આ વિધેય ત્યાર પછી કિંમતોની બાદબાકી કરી `result` નામના ચલમાં પરિણામનો સંગ્રહ કરે છે. ત્યાર બાદ `result` ચલની કિંમત `main()` વિધેયના વિધાનને પરત કરવામાં આવે છે અને `difference` નામના ચલમાં તેનો સંગ્રહ કરવામાં આવે છે. અંતમાં `main()` વિધેય પરિણામને સ્ક્રીન પર દર્શાવી પ્રોગ્રામ પૂરો કરે છે. ઉદાહરણ 10.3નું પરિણામ આકૃતિ 10.6માં દર્શાવ્યું છે.

```
harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ ./a.out

Enter the value of first: 55

Enter the value of second: 24

The value of 55 - 24 is 31

harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$
```

આકૃતિ 10.6 : ઉદાહરણ 10.3નું પરિણામ

જો કે આકૃતિ 10.4માં દર્શાવેલું સી પ્રોગ્રામનું માળખું તમામ પ્રોગ્રામરો દ્વારા ઉપયોગમાં લેવામાં આવતું પ્રમાણભૂત માળખું નથી. આકૃતિ 10.1માં દર્શાવ્યા મુજબ એ પણ શક્ય છે કે આમાંના કેટલાક વિભાગો પ્રોગ્રામમાં ઉપયોગમાં લેવાયા ન હોય તથા કેટલાક વિભાગોના સ્થાનની અદલાબદલી પણ શક્ય છે. જેમ કે, કોડ લિસ્ટિંગ 10.1માં દર્શાવ્યું છે. તે મુજબ ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત વિધેયના વિભાગને `main()` વિધેયના વિભાગ સાથે બદલવામાં આવ્યો છે. અહીં ઉપયોગકર્તા દ્વારા નિર્મિત વિધેયને `main()` વિધેયની પહેલા ઉમેરવામાં આવ્યું છે.

```
/* Example 4 : Program to subtract two numbers */

#include <stdio.h>

/* Beginning of user defined functions other than main( ) */
int subtract(int first, int second)
{
    int result;
    result = first - second;
    return(result);
}
/* End of function subtract( ) */
```



```

/* Beginning of function main() */
int main ( )
{
    int first, second, difference;
    printf("\nEnter the value of first: ");
    scanf("%d", &first);
    printf("\nEnter the value of second: ");
    scanf("%d", &second);
    difference = subtract(first, second);
    printf("\nThe value of %d - %d is %d \n \n", first, second, difference);
    return;
}
/* End of main program */

```

કોડ-લિસ્ટિંગ 10.1 : ઉદાહરણ 10.4નો કોડ

બે તફાવતોને બાદ કરતાં આ પ્રોગ્રામ ઉદાહરણ 10.3ને સમાન છે. પ્રથમ તફાવત એ છે કે subtract() વિધેયને main() વિધેયની પહેલાં વ્યાખ્યાયિત કરવામાં આવ્યું છે અને બીજું એ કે પ્રોટોટાઈપ વિધાનનો ઉપયોગ કરવામાં આવ્યો નથી. ઉદાહરણ 10.3માં difference = subtract (first, second); વિધાન દ્વારા નિયંત્રણ પ્રવાહ main()ની નીચેની તરફ મોકલવામાં આવે છે જ્યારે ઉદાહરણ 10.4માં નિયંત્રણ પ્રવાહ main()ની ઉપરની તરફ જાય છે.

સી ભાષાનાં મૂળાક્ષર (C Character set)

તમને યાદ હશે કે જ્યારે આપણે શાળામાં કોઈ નવી ભાષાનો અભ્યાસ કરતા હતા ત્યારે સૌપ્રથમ આપણે તેના મૂળાક્ષરો શીખ્યા હતા. કોઈ પણ ભાષામાં મૂળાક્ષરો શબ્દોના બંધારણમાં સહાયક બને છે. સી ભાષાને પણ પોતાના મૂળાક્ષરોનો ગણ (character set) છે. આ મૂળાક્ષરોને ચાર વર્ગોમાં વહેંચી શકાય :

1. અક્ષરો (Letters)
2. અંકો (Digits)
3. વ્હાઈટ સ્પેસ (White space)
4. વિશિષ્ટ ચિહ્નો (Special characters)

આ વર્ગોમાં આવતા મૂળાક્ષરો કોષ્ટક 10.1માં દર્શાવ્યા છે :

Letters	Digits	White spaces
A to Z a to z	0 to 9	Blank Space
		Form feed
		Horizontal tab
		New line
		Vertical tab

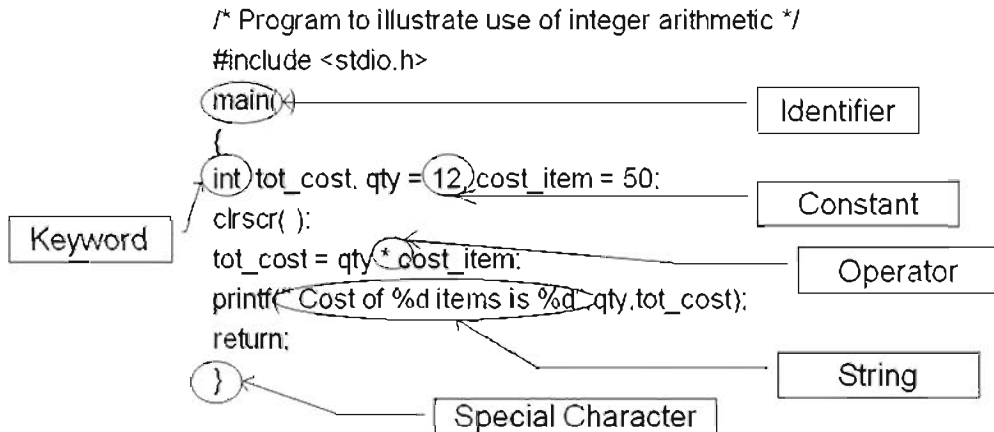
Special Characters			
Operator	Use	Operator	Use
&	Ampersand	>	Greater than
'	Apostrophe	#	Hash sign
*	Asterisk	<	Less than
@	At the rate	-	Minus
\	Backward slash	()	Parenthesis left / right
{ }	Braces left / right	%	Per cent
[]	Bracket left / right	.	Period
^	Caret	+	Plus
:	Colon	?	Question mark
,	Comma	"	Quotation mark
\$	Dollar	;	Semi colon
=	Equal to	~	Tilde
!	Exclamation	_	Under score
/	Forward slash		Vertical bar

કોષ્ટક 10.1 : સી ભાષાના મૂળાક્ષરો

કોષ્ટક 10.1 માં દર્શાવ્યા મુજબ આપણે અંગ્રેજી ભાષાના કેટલાક અક્ષરોનો ઉપયોગ કર્યો છે. આથી હવે આપણે આ શબ્દોનો સી ભાષાની વાક્યરચનામાં ઉપયોગ કરતાં શીખીશું.

સી શબ્દો (C Words)

કોષ્ટક 10.1માં દર્શાવેલા મૂળાક્ષરોનો ઉપયોગ કરી સી ભાષામાં શબ્દોની રચના કરવામાં આવે છે. આ શબ્દોનો ઉપયોગ કરી સી વિધાન બનાવવામાં આવે છે. આમ, તાર્કિક રીતે ક્રમબદ્ધ લખવામાં આવેલાં સી વિધાનોના સમૂહને સી પ્રોગ્રામ તરીકે ઓળખવામાં આવે છે. સી ભાષામાં આવેલા શબ્દને ટોકન (token) કહે છે. કોષ્ટક 10.1માં આપવામાં આવેલો દરેક અક્ષર પોતે એક ટોકન છે. મૂળભૂત રીતે સી ભાષામાં છ પ્રકારના ટોકન આપવામાં આવે છે : કી-વર્ડ, આઈડેન્ટિફાયર, અચળ, સ્ટ્રિંગ, ઓપરેટર અને વિશિષ્ટ ચિહ્નો. સી પ્રોગ્રામમાં આ ટોકનનો ઉપયોગ આકૃતિ 10.7માં દર્શાવ્યો છે.



આકૃતિ 10.7 : સી પ્રોગ્રામમાં શબ્દો (token)નો ઉપયોગ

કી-વર્ડ (Key word)

આપણે સૌએ ક્યારેક તો શબ્દકોશનો ઉપયોગ કર્યો જ છે. જે-તે ભાષામાં આવેલા કોઈ પૂર્વવ્યાખ્યાયિત શબ્દનો અર્થ શોધવા માટે આપણે શબ્દકોશનો ઉપયોગ કરીએ છીએ. સી ભાષા પણ આવા પૂર્વવ્યાખ્યાયિત શબ્દો ધરાવે છે. ચોક્કસપણે કહીએ તો ANSI C ધારાધોરણ 32 પૂર્વવ્યાખ્યાયિત શબ્દોને સમર્થન આપે છે. સી ભાષામાં આપેલા આ પૂર્વવ્યાખ્યાયિત શબ્દોને કી-વર્ડ તરીકે ઓળખવામાં આવે છે. દરેક કી-વર્ડ સાથે એક સુનિશ્ચિત અર્થ સંકળાયેલો છે. ANSI Cમાં ઉપલબ્ધ કી-વર્ડની યાદી કોષ્ટક 10.2માં આપવામાં આવી છે.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

કોષ્ટક 10.2 : ANSI સી કી-વર્ડ

અભ્યાસક્રમના હવે પછીના પ્રકરણમાં આ કી-વર્ડ અને તેના ઉપયોગો વિશે ચર્ચા કરવામાં આવી છે.

આઈડેન્ટિફાયર (Identifier)

ઉપયોગકર્તા દ્વારા સી મૂળાક્ષરોનો ઉપયોગ કરીને બનાવવામાં આવેલ શબ્દોને આઈડેન્ટિફાયર કહે છે. તેમાં અક્ષરો, અંકો અને કેટલાંક વિશિષ્ટ ચિહ્નોનો સમાવેશ કરી શકાય છે. "difference", "main()", "PI", "float" વગેરે આઈડેન્ટિફાયરનાં કેટલાંક ઉદાહરણ છે. અહીં "difference" ચલનું નામ છે, "main()" વિધેયનું નામ છે, "PI" સાંકેતિક અચળનું નામ છે અને "float" એ પૂર્વવ્યાખ્યાયિત શબ્દ છે. હવે, આગળ વધતાં પહેલાં 'ચલ' શું છે તે વિશે જોઈએ.

ચલ (Variable)

સી પ્રોગ્રામ સામાન્ય રીતે ઇનપુટ સ્વીકારે છે, આ ઇનપુટ ડેટાના સ્વરૂપમાં આવે છે. ડેટાનો સંગ્રહ અને ઉપયોગ કરવા માટે આપણે મેમરીની જગ્યાનો ઉપયોગ કરીએ છીએ. મેમરીમાં આપેલ ડેટા તેના પર કરવામાં આવતી પ્રક્રિયાને આધારે બદલાય છે. મેમરી જગ્યામાં આપેલ ડેટાને ચલ (variable) તરીકે ઓળખાતા એક નામ દ્વારા નિર્દેશિત કરવામાં આવે છે. પોતાને બદલી શકવાની ક્ષમતા ધરાવતા ડેટાને ચલ કહે છે. સી ભાષામાં ચલનું નામ વ્યાખ્યાયિત કરવા કેટલાક નિયમોનું પાલન કરવું જરૂરી છે. આ નિયમો નીચે દર્શાવ્યા છે :

1. ચલનું નામ કી-વર્ડના નામ જેવું ન હોવું જોઈએ.
2. ચલના નામમાં મૂળાક્ષરો, અંકો અને અંડરસ્કોરનો ઉપયોગ કરી શકાય છે. અન્ય કોઈ સ્પેશિયલ કેરેક્ટર માન્ય નથી.
3. ચલના નામનો પ્રથમ અક્ષર મૂળાક્ષર અથવા અંડરસ્કોર હોય તે જરૂરી છે.
4. ANSI ધારાધોરણ પ્રમાણે ચલના નામની મહત્તમ લંબાઈ 31 અક્ષર છે. જો કે, તે કમ્પાઈલર પર આધારિત છે.
5. ચલના નામ કેસ-સેન્સિટિવ છે, એટલે કે num, nuM, nUM, nUm અને Num આ તમામ જુદા જુદા

ચલનાં કેટલાંક માન્ય અને અમાન્ય નામ કોષ્ટક 10.3માં દર્શાવ્યાં છે :

માન્ય
Double → Double એ કી-વર્ડ doubleને સમાન નથી.
INTEREST → કેપિટલ અક્ષરો સી મૂળાક્ષરોનો ભાગ છે.
total_quantity → અન્ડરસ્કોર અને અક્ષરો માન્ય છે.
mark100 → પ્રથમ અક્ષર આલ્ફાબેટ હોવાથી માન્ય છે.
_file → પ્રથમ અક્ષર અન્ડરસ્કોર તરીકે શક્ય છે.
અમાન્ય
char → કી-વર્ડનો ઉપયોગ માન્ય નથી.
Total Value → ખાલી જગ્યા માન્ય નથી.
total&value → વિશિષ્ટ અક્ષરો માન્ય નથી.
10mark → પ્રથમ અક્ષર અન્ડરસ્કોર કે આલ્ફાબેટ જોઈએ.

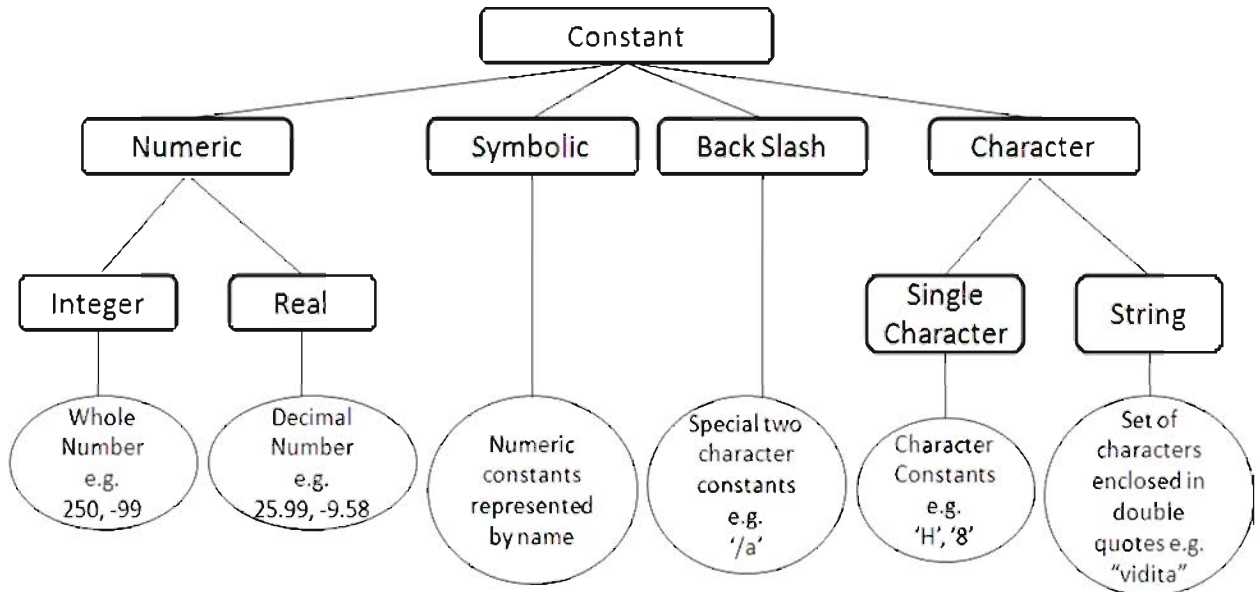
કોષ્ટક 10.3 : સી ચલનાં ઉદાહરણ

અચળ (Constant)

પ્રોગ્રામના અમલીકરણ દરમિયાન જે ઘટકની કિંમત બદલી શકાતી નથી તેને અચળ તરીકે ઓળખવામાં આવે છે. સી અચળોને આકૃતિ 10.7માં દર્શાવ્યા મુજબ જુદા જુદા વર્ગોમાં વહેંચી શકાય છે. આ અચળોની વિસ્તૃત ચર્ચા કરીએ.

સાંખ્યિક અચળ (Numeric constant)

આંકડાકીય વિગતોનો સંગ્રહ કરતા અચળને સાંખ્યિક અચળ કહે છે. સાંખ્યિક અચળોને બે વર્ગોમાં વિભાજિત કરી શકાય : પૂર્ણાંક (integer) અચળ અને અપૂર્ણાંક (real) અચળ



આકૃતિ 10.7 : સી અચળો

પૂર્ણાંક અચળ (Integer constant)

પૂર્ણાંક અચળ એટલે દશાંશ ચિહ્ન વગરની સંખ્યાઓ (whole numbers). પૂર્ણાંક અચળો ત્રણ પ્રકારના છે : દશાંકી (decimal), સોળાંકી (hexadecimal) અને અષ્ટાંકી (octal). જુદી જુદી અંક-પદ્ધતિઓનાં ઉદાહરણ પરિશિષ્ટ I માં આપવામાં આવ્યાં છે.

દશાંકી અચળો 0 થી 9 અંકોનો ઉપયોગ કરે છે. આ અંકોની આગળ યુનરી ધન (plus) કે ઋણ (minus) નિશાની ઉમેરી શકાય છે.

સોળાંકી અચળો 0 થી 9 અંકો અને A થી F અક્ષરોનો ઉપયોગ કરે છે. અહીં A થી F અક્ષરો અનુક્રમે 10 થી 15 સંખ્યાઓનો નિર્દેશ કરે છે. સોળાંકી સંખ્યાનો આધાર 16 છે. આ કિંમતોનો સી ભાષામાં ઉપયોગ કરવામાં આવે ત્યારે તેને દશાંકી સંખ્યાથી અલગ પાડવા તે સંખ્યાની આગળ 0x અથવા 0X ઉમેરવામાં આવે છે.

અષ્ટાંકી અચળો 0 થી 7 અંકોનો ઉપયોગ કરે છે. આ અંક-પદ્ધતિનો આધાર 8 છે. આ કિંમતોનો સી ભાષામાં ઉપયોગ કરવામાં આવે ત્યારે દશાંકી સંખ્યાથી અલગ પાડવા તેની આગળ 0 ઉમેરવામાં આવે છે. કોષ્ટક 10.4 કેટલાક માન્ય અને અમાન્ય પૂર્ણાંક અચળ દર્શાવે છે.

માન્ય
40000 → માન્ય દશાંકી ધન પૂર્ણાંક સંખ્યા
-120 → માન્ય દશાંકી ઋણ પૂર્ણાંક સંખ્યા
79999L → માન્ય દશાંકી ધન લોંગ અપૂર્ણાંક સંખ્યા
0xB5 → માન્ય સોળાંકી સંખ્યા
0X79 → માન્ય સોળાંકી સંખ્યા
045 → માન્ય અષ્ટાંકી સંખ્યા
અમાન્ય
25,000 → અલ્પવિરામ માન્ય નથી.
-100.0 → તે અપૂર્ણ અચળ છે.
Rs79999 → અક્ષરો માન્ય નથી.
0xG → G સોળાંકી અંકમાં ન હોઈ શકે.
0X8,9 → અલ્પવિરામ માન્ય નથી.
096 → 9 અષ્ટાંકી સંખ્યાના ભાગ સ્વરૂપે ન હોઈ શકે.

કોષ્ટક 10.4 : પૂર્ણાંક અચળ

અપૂર્ણ અચળ (Real constant)

અપૂર્ણાંક ભાગ ધરાવતા દશાંકી અંક અપૂર્ણ અચળ તરીકે ઓળખાય છે. 10.50, -65.75 વગેરે અપૂર્ણ અચળનાં ઉદાહરણ છે. અપૂર્ણ અચળોની રજૂઆત વૈજ્ઞાનિક સ્વરૂપે પણ કરી શકાય છે. આ પ્રકારની રજૂઆત માટે મેન્ટિસા (mantissa) અને એક્સ્પોનન્ટ (exponent)નો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, 25.75 સંખ્યાને વૈજ્ઞાનિક સ્વરૂપમાં 0.2575e2 અથવા 0.2575E2 તરીકે રજૂ કરી શકાય છે. અહીં 0.2575 ને મેન્ટિસા અને 2ને એક્સ્પોનન્ટ કહે છે. વળી, "e" અને "E" સરખા છે. કોષ્ટક 10.5માં કેટલાક માન્ય અને અમાન્ય અપૂર્ણ અચળની યાદી આપી છે.

માન્ય
250.00 → માન્ય દશાંકી અપૂર્ણાંક સંખ્યા
295 → માન્ય દશાંકી સંખ્યા, આપોઆપ 295.00 માં રૂપાંતરણ પામશે.
-15.55 → માન્ય ઋણ દશાંકી સંખ્યા
-20.5e5 → માન્ય વૈજ્ઞાનિક કિંમત
15E-2 → માન્ય વૈજ્ઞાનિક કિંમત
અમાન્ય
19,800.00 → અલ્પવિરામ માન્ય નથી.
\$786 → વિશિષ્ટ ચિહ્ન માન્ય નથી.
-9.4e5.0 → એક્સ્પોનન્ટ પૂર્ણાંક હોવો જોઈએ.
51E -2 → ખાલી જગ્યા માન્ય નથી.

કોષ્ટક 10.5 : અપૂર્ણ અચળ

અક્ષર પ્રકારના અચળ (Character constant)

નામ પ્રમાણે આ અચળમાં અક્ષરોનો સમાવેશ કરવામાં આવે છે. સી ભાષામાં બે પ્રકારના અક્ષર અચળો ઉપલબ્ધ છે : એક અક્ષર સ્વરૂપે અચળ અને સ્ટ્રિંગ અચળ

એક અક્ષર સ્વરૂપે અચળ (Single character constant)

આ પ્રકારના અચળમાં એક અક્ષરને એક અવતરણ ચિહ્ન (single quote) માં આવરીને રજૂ કરવામાં આવે છે. 'H', 'v', '7', '\$' વગેરે એક અક્ષર સ્વરૂપે રજૂ કરવામાં આવેલા અચળનાં કેટલાંક ઉદાહરણ છે. અહીં દરેક અક્ષર અચળ ASCII (American Standard Code for Information Interchange) નામે ઓળખાતી કિંમત સાથે જોડાયેલ હોય છે. ASCII કિંમતો અને તેની સાથે જોડાયેલા જુદા જુદા અક્ષરોની વિગતો પરિશિષ્ટ II માં આપવામાં આવી છે.

સ્ટ્રિંગ અચળ (String constant)

સ્ટ્રિંગ અચળને બે અવતરણ ચિહ્નો(double quotes)માં આવરીને લખવામાં આવેલા અક્ષરોના સમૂહ દ્વારા રજૂ કરવામાં આવે છે. "C Language", "Bye", "V" વગેરે સ્ટ્રિંગ અચળનાં કેટલાંક ઉદાહરણ છે. સ્ટ્રિંગ અચળ સાથે કોઈ ASCII કિંમત જોડાયેલી હોતી નથી. અહીં, સ્ટ્રિંગ અચળ "V" અને અક્ષર સ્વરૂપે રહેલ અચળ 'V' સરખાં નથી. સ્ટ્રિંગ "V" માટે બે મેમરી-સ્થાન આપવામાં આવે છે, જ્યારે 'V' અક્ષરને માત્ર એક મેમરી-સ્થાન આપવામાં આવે છે. કારણ કે, સી ભાષામાં સ્ટ્રિંગનો અંત નલ અક્ષર '\0' થી થાય છે.

બેક સ્લેશ અક્ષરો (Back slash characters)

નામ અનુસાર 'સિંગલ કેરેક્ટર અચળ' એક અક્ષરનો ઉપયોગ કરે છે જ્યારે સ્ટ્રિંગ અનેક અક્ષરોનો ઉપયોગ કરે છે. સી ભાષા એક અન્ય પ્રકારના વિશિષ્ટ અચળ પૂરાં પાડે છે જે બે અક્ષરોનો ઉપયોગ કરે છે. આ અચળોને બેક સ્લેશ અક્ષરો (back slash characters) અથવા એસ્કેપ સીકવન્સ (escape sequence) કહે છે. પ્રથમ અક્ષર હંમેશા બેક સ્લેશ "\" હોવાને કારણે આ અચળોને બેક સ્લેશ અક્ષરો તરીકે ઓળખવામાં આવે છે તથા આ અચળોના પરિણામસ્વરૂપે વ્હાઈટ સ્પેસ દર્શાવાતી હોવાને કારણે તેને 'એસ્કેપ સીકવન્સ' પણ કહે છે. સિંગલ કેરેક્ટર અચળની જેમ આ અચળોની સાથે પણ એક ASCII કિંમત સંકળાયેલી હોય છે. સી ભાષામાં ઉપલબ્ધ બેક સ્લેશ અચળો અને તેના ઉપયોગો તથા ASCII કિંમતોની યાદી કોષ્ટક 10.6માં આપવામાં આવી છે.

Back Slash Character	Use	ASCII Value
\0	નલ કિંમત ઉમેરવા માટે	0
\a	શ્રવણીય એલર્ટ ઉમેરવા માટે	7
\b	બેકસ્પેસ ઉમેરવા માટે	8
\t	આડી ટેબ ઉમેરવા માટે	9
\n	નવી લીટી ઉમેરવા માટે	10
\v	ઊભી ટેબ ઉમેરવા માટે	11
\f	ફોર્મ ફીડ ઉમેરવા માટે	12
\r	કેરેજ રીટર્ન ઉમેરવા માટે	13
\"	બે અવતરણ ચિહ્ન ઉમેરવા માટે	34
\'	એક અવતરણ ચિહ્ન ઉમેરવા માટે	39
\?	પ્રશ્નાર્થ ચિહ્ન ઉમેરવા માટે	63
\\	બેક સ્લેશ ઉમેરવા માટે	92

કોષ્ટક 10.6 બેક સ્લેશ અક્ષરો

એસ્કેપ સિક્વન્સનો ઉપયોગ મુખ્યત્વે ગોઠવણી(formatting)ના હેતુસર કરવામાં આવે છે. ઉદાહરણ તરીકે '\n' નો ઉપયોગ કરી ઈનપુટ કે આઉટપુટ સમયે નવી લીટી ઉમેરી શકાય છે. આ અક્ષરોનો ઉપયોગ તમે પુસ્તકમાં અનેક જગ્યાએ જોઈ શકશો.

સાંકેતિક અચળ (Symbolic Constant)

સાંકેતિક આઈડેન્ટિફાયરની મદદથી ઘણા આંકડાકીય અને અક્ષરીય અચળોને વ્યાખ્યાયિત કરી શકાય છે. આ પ્રકારના અચળને સાંકેતિક અચળ કહે છે. સાંકેતિક અચળોને વ્યાખ્યાયિત કરવા માટે નીચે જણાવેલ વાક્યરચનાનો ઉપયોગ કરવામાં આવે છે :

#define identifier value

અહીં, #defineને પ્રી-પ્રોસેસર ડાઈરેક્ટિવ (pre-processor directive) તરીકે ઓળખવામાં આવે છે, જે અચળ વ્યાખ્યાયિત કરવાનો છે તેનું સાંકેતિક નામ 'આઈડેન્ટિફાયર' છે અને value એ સ્વયં અચળ છે. સાંકેતિક અચળનાં કેટલાંક ઉદાહરણ નીચે દર્શાવેલાં છે :

```
#define PI 3.14
```

```
#define MAXVALUE 100
```

```
#define f float
```

અહીં પ્રથમ વિધાન "PI" નામના સાંકેતિક અચળને વ્યાખ્યાયિત કરે છે જેની કિંમત અપૂર્ણાંક "3.14" છે. બીજું વિધાન "MAXVALUE" નામના સાંકેતિક અચળને વ્યાખ્યાયિત કરે છે જેની કિંમત પૂર્ણાંક 100 છે અને અંતિમ વિધાન "f" નામના સાંકેતિક અચળને વ્યાખ્યાયિત કરે છે, જેની કિંમત સી ભાષામાં આવેલ કી-વર્ડ "float" છે.

પ્રોગ્રામમાં ઉપયોગમાં લેવામાં આવેલા તમામ સાંકેતિક અચળોને તેની વ્યાખ્યામાં આપેલ કિંમત સાથે બદલવાની સૂચના પ્રી-પ્રોસેસર ડાઈરેક્ટિવ વિધાન કંપાઈલરને આપે છે. એક ગોલક (Sphere)ની સપાટીનું ક્ષેત્રફળ શોધવા માટેનો પ્રોગ્રામ ઉદાહરણ 10.5માં દર્શાવ્યો છે, જે પ્રી-પ્રોસેસર ડાઈરેક્ટિવનો ઉપયોગ કરે છે. ઉદાહરણ 10.5નું કોડ લિસ્ટિંગ આકૃતિ 10.8માં દર્શાવ્યું છે તથા આકૃતિ 10.9 પ્રોગ્રામનું પરિણામ દર્શાવે છે.

```
10_5.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 10_5.c
/* Example 5: Program to find surface area of a sphere */

#include <stdio.h>

/* Definition of a symbolic constant */

#define F float
#define P printf
#define S scanf
#define PI 3.14

int main( )
- {
    F radius, sarea;
    P("\nEnter the value of radius: ");
    S("%f", &radius);
    sarea = 4 * PI * radius * radius;
    P("\nSurface Area of sphere with radius %.2f is %.2f\n", radius, sarea);
    return 0;
}
/* End of program */
```

આકૃતિ 10.8 : ઉદાહરણ 10.5નું કોડ વિસ્તૃત

```
harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ ./a.out
Enter the value of radius: 2.5
Surface Area of sphere with radius 2.50 is 78.50
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$
```

આકૃતિ 10.9 : ઉદાહરણ 10.5નું પરિણામ

સમજૂતી

આ પ્રોગ્રામમાં ચાર સાંકેતિક અચળો વ્યાખ્યાયિત કર્યા છે : 'F' કી-વર્ડ 'float' માટે, 'P' વિધેય 'printf' માટે, 'S' વિધેય 'scanf' માટે તથા 'PI' અપૂર્ણ ક્રિમત '3.14' માટે. main() વિધેયનું પ્રથમ વિધાન F ડેટા પ્રકાર માટે બે ચલ વ્યાખ્યાયિત કરે છે. F એ floatનો નિર્દેશ કરતું હોવાથી ખરેખર અહીં બે float ચલને વ્યાખ્યાયિત કરવામાં આવે છે. ત્રીજું વિધાન Pનો ઉપયોગ કરી સંદેશ દર્શાવે છે. ત્યારપછી Sનો ઉપયોગ કરી ત્રિજ્યાની ક્રિમત વાંચવામાં આવે છે અને ગોલક (Sphere)ના સપાટીનાં ક્ષેત્રફળની ગણતરી કરવામાં આવે છે. અંતમાં ફરી Pનો ઉપયોગ કરી સંદેશ સાથે સપાટીનું ક્ષેત્રફળ દર્શાવવામાં આવે છે.

સી પ્રોગ્રામમાં યાદ રાખવાના મુદ્દા (Points to be remember in C program)

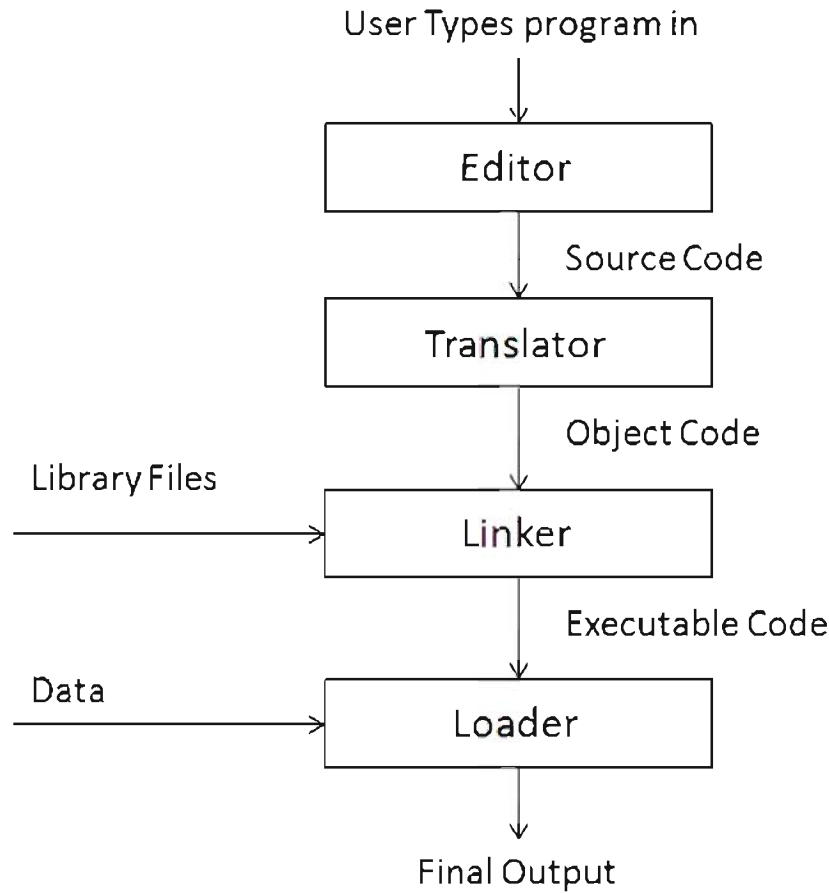
સી ભાષામાં પ્રોગ્રામિંગ સરળ હોવા છતાં, તેનો મહાવરો જરૂરી છે. તમામ સી પ્રોગ્રામરોએ નીચેના વિભાગમાં આપવામાં આવેલ મુદ્દા યાદ રાખવા જોઈએ :

- હેડર ફાઈલને main() વિધેયની પહેલાં ઉમેરવી જોઈએ.
- અમલ કરી શકાય તેવા કોઈ પણ સી પ્રોગ્રામમાં હંમેશા main() વિધેય હોય છે.

- સી પ્રોગ્રામનું અમલીકરણ main() પછીના ઉઘડતા છગડિયા કૌંસ ({})થી શરૂ કરવામાં આવે છે.
- સી પ્રોગ્રામમાં સરખાં નામ ધરાવતાં બે વિષય માન્ય નથી.
- સામાન્ય રીતે, સી પ્રોગ્રામ નાના (small) અક્ષરો દ્વારા લખવામાં આવે છે. આઈડેન્ટિફાયર કેસ-સેન્સિટિવ છે.
- સી પ્રોગ્રામના બે શબ્દો વચ્ચે ઓછામાં ઓછી એક ખાલી જગ્યા હોવી જરૂરી છે.
- સામાન્ય રીતે સી ભાષામાં વિધાનોને અર્ધવિરામ (semicolon) દ્વારા પૂરાં કરવામાં આવે છે.
- ખાલી જગ્યા ઉમેરી શકાય તેવા તમામ સ્થાને નોંધ (comment) ઉમેરી શકાય છે.
- દરેક ઉઘડતા છગડિયા કૌંસ({})ને સંબંધિત પૂરો થતો છગડિયો કૌંસ (}) હોવો અનિવાર્ય છે.

સી પ્રોગ્રામનું અમલીકરણ (Execution of C program)

અત્યાર સુધીમાં આપણે ઘણા સી પ્રોગ્રામ અને તેનાં પરિણામ જોયાં. હવે આપણે સી પ્રોગ્રામનો અમલ કરતાં શીખીએ. પ્રત્યક્ષ અનુભવ કરતાં પહેલાં આ માટે જરૂરી એવાં સોપાન સમજીએ. આકૃતિ 10.9(a)માં આ સંપૂર્ણ પ્રક્રિયા દર્શાવી છે.

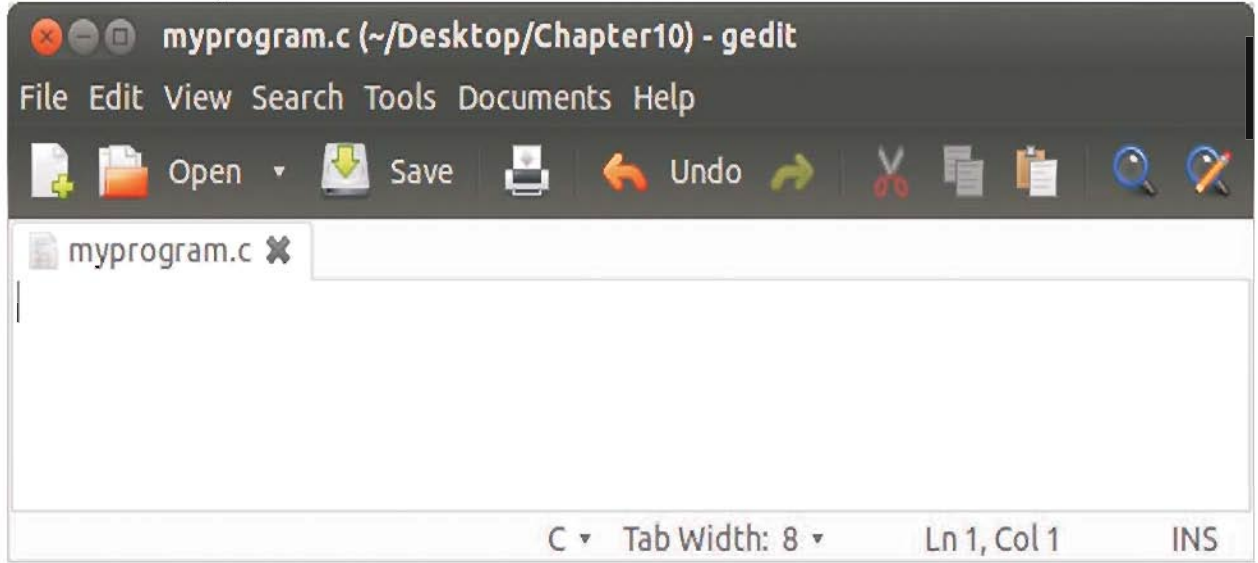


આકૃતિ 10.9 (a) : સી પ્રોગ્રામનાં અમલીકરણનાં પગલાં

આકૃતિ 10.9(a) માં દર્શાવ્યા મુજબ, ટેક્સ્ટ એડિટરનો ઉપયોગ કરી પ્રોગ્રામ લખવો એ સૌ પ્રથમ સોપાન છે. ટેક્સ્ટ એડિટરની મદદથી લખવામાં આવેલા પ્રોગ્રામને સોર્સ કોડ (source code) અથવા સોર્સ પ્રોગ્રામ (source program) કહે છે. સી ભાષામાં લખવામાં આવેલ સી ફાઈલનું અનુલંબન ".c" છે. ત્યાર પછી કંપાઈલરની મદદથી પ્રોગ્રામને સંકલિત (compile) કરવામાં આવે છે. કંપાઈલર એ એક અનુવાદક (translator) છે, જે સોર્સ પ્રોગ્રામને ઓબ્જેક્ટ કોડ (object code) કે ઓબ્જેક્ટ પ્રોગ્રામ (object program) નામે ઓળખાતા મશીન ભાષાના કોડમાં રૂપાંતરિત કરે છે. ઓબ્જેક્ટ કોડના જુદાં જુદાં સ્વરૂપો ઉપલબ્ધ છે. ત્યાર પછી લિંકર (linker) નામે ઓળખાતા પ્રોગ્રામની મદદથી ઓબ્જેક્ટ કોડ સાથે લાઈબ્રેરી ફાઈલનું જોડાણ (linking) કરી એક્ઝિક્યુટેબલ પ્રોગ્રામ (executable program) કે એક્ઝિક્યુટેબલ કોડ (executable code) તૈયાર કરવામાં આવે છે. અંતમાં, પરિણામ દર્શાવવા માટે આ એક્ઝિક્યુટેબલ કોડને જરૂરી વગતો સાથે લોડર (loader) નામના પ્રોગ્રામની મદદથી મેમરીમાં મૂકવામાં આવે છે.

આ પુસ્તકમાં આપણે gcc નામના કંપાઈલરનો ઉપયોગ કર્યો છે. gcc કંપાઈલર Free Software Foundation દ્વારા પૂરું પાડવામાં આવ્યું છે. તે યુનિક્સ / લિનક્સ આધારિત ANSI સી કમ્પાઈલર છે. સામાન્ય રીતે તેને કમાન્ડ લાઈન દ્વારા ઉપયોગમાં લેવામાં આવે છે, પરંતુ તેને SciTE જેવા ટેક્સ્ટ એડિટર અથવા Eclipse જેવા ઇન્ટિગ્રેટેડ ડેવલપમેન્ટ એન્વાયર્નમેન્ટ (Integrated Development Environment - IDE) સાથે પણ સાંકળી શકાય છે. મોટાભાગના તમામ લિનક્સ-પ્રસ્થાપનો સાથે તે પૂર્વસ્થાપિત હોય છે, જેથી આપણે તેનો ઉપયોગ કોઈ જ મુશ્કેલી વગર સરળતાથી કરી શકીએ છીએ.

ચાલો, આપણે કમ્પાઈલરનો ઉપયોગ કરીએ. સૌ પ્રથમ ટેક્સ્ટ એડિટરનો ઉપયોગ કરી પ્રોગ્રામ લખવાની જરૂર પડશે. સી પ્રોગ્રામ લખવા માટે કોઈ પણ ટેક્સ્ટ એડિટરનો ઉપયોગ કરી શકાય. લિનક્સ ઓપરેટિંગ સિસ્ટમમાં vi, gedit, emacs અને બીજા ઘણા ટેક્સ્ટ એડિટર ઉપલબ્ધ છે. તમને અનુકૂળ હોય તેવું કોઈ એક ટેક્સ્ટ એડિટર પસંદ કરો. એકમાત્ર વાત અહીં ધ્યાનમાં રાખવી જોઈએ કે બનાવવામાં આવેલી ફાઈલનો .c અનુલંબન આપી સંગ્રહ કરવો જરૂરી છે. અક્ષર 'c' નાના (small) અક્ષરોમાં લખાયેલ હોય તે પણ જરૂરી છે. આપણે સી પ્રોગ્રામની રચના કરવા માટે gedit નામના ટેક્સ્ટ એડિટરનો ઉપયોગ કરીએ. ટર્મિનલ ખોલી, તેના પ્રોમ્પ્ટ પર gedit myprogram.c ટાઈપ કરો. આમ કરવાથી આકૃતિ 10.10માં દર્શાવ્યા મુજબનું એક ખાલી એડિટર ખુલશે.



આકૃતિ 10.10 : ખાલી એડિટર સ્ક્રીન

આ ખાલી એડિટર સ્ક્રીનમાં હવે કોડ લિસ્ટિંગ 10.2માં આપવામાં આવેલ વિગતો ઉમેરો.

```
/* Example 6 : My C program */

#include <stdio.h>

int main()
{
    printf("\nWelcome to the world of C programming using gcc\n");
    return 0;
}
```

કોડ લિસ્ટિંગ 10.2 : ઉદાહરણ 10.6નો કોડ

પ્રોગ્રામ તૈયાર થઈ ગયા પછી એડિટર આકૃતિ 10.11માં દર્શાવ્યા મુજબ દેખાશે. ફાઈલનો સંગ્રહ કરી એડિટર બંધ કરો.

```

myprogram.c (~/Desktop/Chapter10) - gedit
File Edit View Search Tools Documents Help
Open Save Print Undo
myprogram.c
/* Example 6: My C program */
#include <stdio.h>
int main()
{
    printf("\nWelcome to the world of C programming using gcc\n");
    return 0;
}
C Tab Width: 8 Ln 8, Col 2 INS

```

આકૃતિ 10.11 : ઉદાહરણ 10.6ના કોડ લિસ્ટિંગ સાથે એડિટર

હવે આપણે તેનું કમ્પાઇલેશન કરવા તૈયાર છીએ. ટર્મિનલમાં પાછા ફરી નીચે જણાવેલ કમાન્ડ ટાઇપ કરો :

```
$ gcc myprogram.c
```

જો પ્રોગ્રામમાં કોઈ ક્ષતિ નહીં હોય તો નવા પ્રોમ્ટની લીટી દર્શાવાશે. આનો અર્થ એ કે કમ્પાઇલેશનની ક્રિયા સફળતાપૂર્વક પૂરી થઈ છે અને સોર્સ ફાઇલનો સંગ્રહ કરવામાં આવ્યો છે તે જ ડિરેક્ટરીમાં a.out પૂર્વ નિર્ધારિત નામ સાથે એક એક્ઝિક્યુટેબલ ફાઇલની રચના કરવામાં આવી છે. જો પ્રોગ્રામમાં ક્ષતિ હશે તો તે દર્શાવતો સંદેશ સ્ક્રીન પર પ્રદર્શિત કરવામાં આવશે. પ્રોગ્રામમાં ક્ષતિ હોય તો તેને દૂર કરી પ્રોગ્રામને પુનઃ કમ્પાઇલ કરવો જરૂરી છે. પ્રોગ્રામનું પરિણામ જોવા માટે નીચે જણાવેલ કમાન્ડ ટાઇપ કરી એન્ટર કી દબાવો :

```
$ ./a.out
```

આ કમાન્ડ દ્વારા myprogram.c પ્રોગ્રામનું પરિણામ આકૃતિ 10.12માં દર્શાવ્યા મુજબ રજૂ કરવામાં આવશે :

```

harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ gedit myprogram.c
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ gcc myprogram.c
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ ./a.out
Welcome to the world of C programming using gcc
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$

```

આકૃતિ 10.12 : ઉદાહરણ 10.6નું પરિણામ

તમામ સી પ્રોગ્રામનું કમ્પાઇલેશન a.out નામની પૂર્વનિર્ધારિત ફાઇલની રચના કરતું હોવા છતાં, પરિણામી ફાઇલનું નામ આપવું એ વધુ સારી ટેવ ગણાય છે. આમ કરવાથી, આપેલ નામ સાથેની પરિણામી ફાઇલ એકવાર બનાવ્યા પછી જ્યાં સુધી મૂળ સોર્સકોડમાં ફેરફાર ન થાય ત્યાં સુધી તેનો અનેક વાર પુનઃઉપયોગ પણ કરી શકાય છે. નીચે જણાવેલ કમાન્ડ પરિણામી ફાઇલને આપવામાં આવતું નામ દર્શાવે છે :

```
$ gcc -o myprogram.o myprogram.c
```

આ કમાન્ડમાં પ્રથમ આર્ગ્યુમેન્ટ myprogram.o પરિણામી ફાઇલનું નામ રજૂ કરે છે. જ્યારે બીજા આર્ગ્યુમેન્ટ સોર્સ ફાઇલનું નામ દર્શાવે છે. પ્રથમ આર્ગ્યુમેન્ટ સાથે .o અનુલંબનનો ઉપયોગ કરવામાં આવ્યો છે તે જુઓ. આ માત્ર નિર્દેશ છે કે આપેલ ફાઇલ પરિણામી (output) ફાઇલ છે. કમ્પાઇલરને આ અનુલંબનની જરૂર નથી, માટે myprogram.oના સ્થાને માત્ર myprogram પણ લખી શકાય છે. આપેલ પ્રોગ્રામનો સફળતાપૂર્વક અમલ થશે તો કમ્પાઇલર હવે a.outને બદલે myprogram.o નામની ફાઇલ બનાવશે. હવે આપણે નીચેના કમાન્ડનો અમલ કરી પ્રોગ્રામનું પરિણામ મેળવી શકીશું :

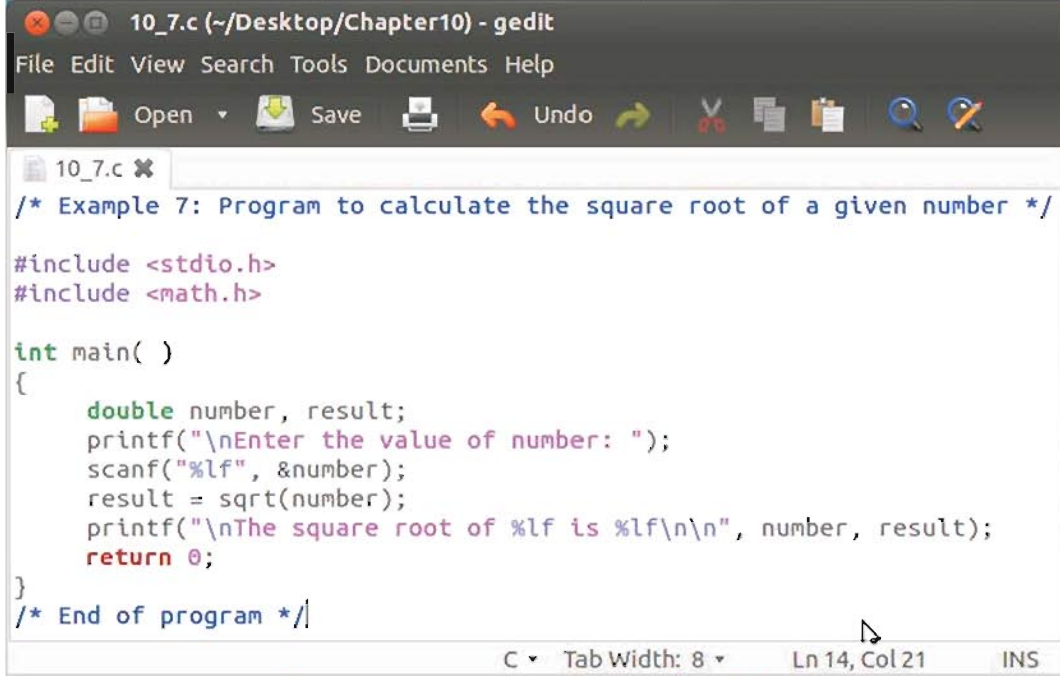
```
$ ./myprogram.o
```

gcc કમાન્ડ સાથે અન્ય ઘણા પ્રાયોનોનો ઉપયોગ પણ કરી શકાય છે. gcc કમાન્ડ વિશેની વિસ્તૃત મદદ મેળવવા માટે નીચે જણાવેલ કમાન્ડનો ઉપયોગ કરી શકાય છે :

\$ man gcc

હવે આપણે અન્ય એક પ્રોગ્રામ લખીને કમ્પાઇલ કરવાનો પ્રયત્ન કરીએ.

ઉદાહરણ 10.7માં દર્શાવેલ પ્રોગ્રામ આંતરપ્રસ્થાપિત વિધેયનો ઉપયોગ કરી ઉપયોગકર્તાએ આપેલ સંખ્યાનું વર્ગમૂળ ગણી આપે છે. આકૃતિ 10.3માં આ ઉદાહરણનું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે.



```
10_7.c (~/Desktop/Chapter10) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
10_7.c x
/* Example 7: Program to calculate the square root of a given number */
#include <stdio.h>
#include <math.h>

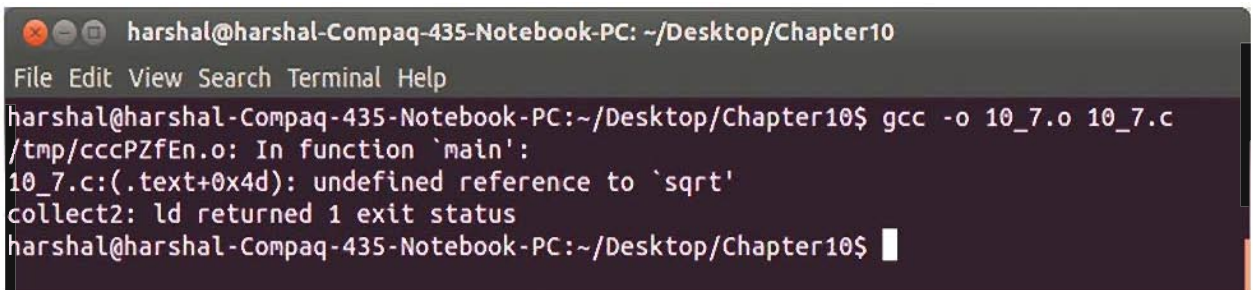
int main( )
{
    double number, result;
    printf("\nEnter the value of number: ");
    scanf("%lf", &number);
    result = sqrt(number);
    printf("\nThe square root of %lf is %lf\n\n", number, result);
    return 0;
}
/* End of program */
C Tab Width: 8 Ln 14, Col 21 INS
```

આકૃતિ 10.13 : ઉદાહરણ 10.7નું કોડ લિસ્ટિંગ

gccનો ઉપયોગ કરી આ પ્રોગ્રામને કંપાઇલ કરવાનો પ્રયત્ન કરીએ. ટર્મિનલ ખોલી કમાન્ડ પ્રોમ્પ્ટ પર નીચે આપેલો કમાન્ડ ટાઇપ કરો :

\$ gcc -o 10_7.o 10_7.c

આ gcc કમાન્ડનું પરિણામ આકૃતિ 10.14માં દર્શાવ્યું છે.



```
harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ gcc -o 10_7.o 10_7.c
/tmp/cccPZfEn.o: In function 'main':
10_7.c:(.text+0x4d): undefined reference to 'sqrt'
collect2: ld returned 1 exit status
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$
```

આકૃતિ 10.14 : ઉદાહરણ 10.7નું પરિણામ

અહીં એ બાબતની નોંધ કરો કે, પરિણામમાં પ્રોમ્પ્ટના સ્થાને "undefined reference to 'sqrt'" સંદેશ દર્શાવવામાં આવ્યો છે. પ્રોગ્રામમાં sqrt() નામના આંતરપ્રસ્થાપિત વિધેયનો ઉપયોગ કર્યો હોવાથી કમ્પાઇલેશન સમયે તેની સાથે ગાણિતિક લાઇબ્રેરી જોડવી જરૂરી છે. નીચે આપેલ કમાન્ડ દ્વારા ગાણિતિક લાઇબ્રેરી જોડી શકાશે.

\$ gcc -o 10_7.o 10_7.c -lm

આ કમાન્ડ આપવાથી પ્રોગ્રામ કોઈ પણ ભૂલના સંદેશ વગર કમ્પાઇલ થઈ જશે. હવે પ્રોમ્પ્ટ પર ./10_7.o ટાઇપ કરી પરિણામ જુઓ. આકૃતિ 10.15માં સાચું પરિણામ દર્શાવ્યું છે.


```

harshal@harshal-Compaq-435-Notebook-PC: ~/Desktop/Chapter10
File Edit View Search Terminal Help
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ gcc -o 10_7.o 10_7.c -lm
harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$ ./10_7.o

Enter the value of number: 25

The square root of 25.000000 is 5.000000

harshal@harshal-Compaq-435-Notebook-PC:~/Desktop/Chapter10$

```

આકૃતિ 10.15 : ઉદાહરણ 10.7નું સાચું પરિણામ

અહીં દર્શાવેલી પદ્ધતિ લિનક્સ ઓપરેટિંગ સિસ્ટમમાં સી પ્રોગ્રામ લખવા અને કમ્પાઇલ કરવા માટેની સૌથી પાયારૂપ પદ્ધતિ છે. અન્ય કોઈ પણ સાદા ટેક્સ્ટ એડિટરનો ઉપયોગ કરવાને બદલે આ પુસ્તકમાં આપણે પ્રોગ્રામ લખવા અને અમલ કરવા માટે SciTE નામના ટેક્સ્ટ એડિટરનો ઉપયોગ કરીશું. SciTE એક જ વિન્ડોમાં પ્રોગ્રામને કંપાઇલ અને અમલ કરવાની સુવિધા પૂરી પાડે છે. ઉપયોગકર્તા પાસેથી ઈનપુટ મેળવવાના હોય તે પ્રકારના પ્રોગ્રામ SciTEમાં લખી શકાય છે, પરંતુ તેનો અમલ કરવા માટે આપણે ટર્મિનલનો ઉપયોગ કરીશું.

તમારા કમ્પ્યુટરમાં યોગ્ય સ્થાન પરથી SciTE ટેક્સ્ટ એડિટર ખોલો. આપણે અગાઉના ઉદાહરણમાં SciTE એડિટર વિન્ડોનો દેખાવ જોયો છે. આકૃતિ 10.1માં દર્શાવ્યા મુજબ SciTE એડિટરની ખાલી સ્ક્રીનમાં ઉદાહરણ 10.1માં આપેલ લખાણ ટાઇપ કરો. ફાઇલને 10_1.c નામ આપી સંગ્રહ કરો. આ માટે Ctrl + S અથવા **File → Save**નો ઉપયોગ કરી શકાય. હવે વિન્ડો આકૃતિ 10.16માં દર્શાવ્યા મુજબના દેખાવ જેવી લાગશે.

```

10_1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 10_1.c

/* Example 1: My first C program */

#include <stdio.h>
int main( )
- {
    printf("Welcome to the world of C programming using Scite \n");
    return 0;
}

```

આકૃતિ 10.16 : SciTE વિન્ડોમાં લખેલ ઉદાહરણ 10.1

એકવાર પ્રોગ્રામ લખાઈ જાય અને સંગ્રહ થઈ જાય પછી તેમાં જો કોઈ વાક્યરચનાની ભૂલ (syntax errors) આવેલી હોય તો તે શોધવાની જરૂર પડશે. આ ભૂલ શોધવા માટે પ્રોગ્રામને કંપાઇલ કરવો પડે. આ માટે Ctrl + F7 કી અથવા Tools → Compile વિકલ્પ પસંદ કરો. જો પ્રોગ્રામમાં કોઈ ક્ષતિ નહીં હોય તો આકૃતિ 10.17માં દર્શાવ્યા મુજબની સ્ક્રીન રજૂ કરવામાં આવશે.

```

10_1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 10_1.c

/* Example 1: My first C program */
#include <stdio.h>
int main( )
- {
    printf("Welcome to the world of C programming using Scite \n");
    return 0;
}

>gcc -pedantic -Os -c 10_1.c -o 10_1.o -std=c99
>Exit code: 0

```

આકૃતિ 10.17 : કમ્પાઇલેશનની સફળતાનો સંદેશ

હવે પ્રોગ્રામનો અમલ કરી શકાશે. આ માટે F5 કી દબાવો અથવા Tools → Go વિકલ્પ પસંદ કરો. આમ કરવાથી આકૃતિ 10.18માં દર્શાવ્યા મુજબ સોર્સકોડની વિન્ડો સાથે આઉટપુટ વિન્ડો દર્શાવવામાં આવશે.

```

10_1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 10_1.c
/* Example 1: My first C program */
#include <stdio.h>
int main( )
{
    printf("Welcome to the world of C programming using Scite \n");
    return 0;
}

>gcc -pedantic -Os -c 10_1.c -o 10_1.o -std=c99
>Exit code: 0
>gcc -pedantic -Os -std=c99 10_1.c -o 10_1
>Exit code: 0
>./10_1
Welcome to the world of C programming using Scite
>Exit code: 0

```

આકૃતિ 10.18 : બે વિન્ડો સાથેનું SciTE એડિટર

અહીં એ જોઈ શકાય છે કે આપણે કરેલી બંને પ્રવૃત્તિઓનું પરિણામ દર્શાવવામાં આવે છે. જ્યારે પ્રોગ્રામને કમ્પાઈલ કરવામાં આવ્યો હતો ત્યારે નીચે આપેલ પરિણામ પ્રથમ દર્શાવવામાં આવ્યું હતું :

```
> gcc -pedantic -Os 10_1.c -o 10_1.o -std=c99
```

```
> Exit code : 0
```

અહીં પરિણામ સ્વરૂપે મળેલી 10_1.o ફાઈલ અમલ થઈ શકે તે પ્રકારની નથી. જ્યારે Goનો ઉપયોગ કરવામાં આવશે ત્યારે નીચે જણાવેલ પરિણામ મેળવી શકીશું :

```
>gcc -pedantic -Os -std=c99 10_1.c -o 10_1
```

```
>Exit code : 0
```

```
>./10_1
```

```
Welcome to the world of C programming using Scite
```

```
>Exit code : 0
```

અહીં ફાઈલ પર બે તબક્કામાં પ્રક્રિયા કરવામાં આવે છે. પ્રથમ તબક્કામાં ફાઈલને કંપાઈલ કરવામાં આવે છે અને 10_1 નામની એક્ઝિક્યુટેબલ ફાઈલ બનાવવામાં આવે છે. આ પ્રક્રિયા પ્રથમ બે લીટી દ્વારા રજૂ કરવામાં આવી છે. બીજા તબક્કામાં (ત્રીજી લીટીમાં દર્શાવ્યા મુજબ) ./10_1 કમાન્ડનો ઉપયોગ કરી આઉટપુટ ફાઈલનો અમલ કરવામાં આવ્યો છે. છેલ્લી બે લીટી પરિણામ દર્શાવે છે.

નોંધ : > નિશાની પછી આવેલું લખાણ એ કમ્પાઈલર દ્વારા કરવામાં આવેલી પ્રક્રિયા છે, જ્યારે ઉપયોગકર્તા માટે સ્ક્રીન પર દર્શાવવાના પરિણામની આગળ > નિશાની ઉમેરવામાં આવતી નથી.

F8 કી દબાવીને અથવા તો **View → Output** વિકલ્પનો ઉપયોગ કરીને આઉટપુટ વિન્ડો દર્શાવવી કે અદૃશ્ય બનાવવી પણ શક્ય છે. અગાઉનાં તમામ પરિણામોને દૂર કરવા Shift + F5 કી અથવા **Tools → Clear Output** વિકલ્પનો ઉપયોગ કરી શકાય છે.

આ ઉદાહરણમાં આપણે એવો પ્રોગ્રામ પસંદ કરેલો જે યોગ્ય રીતે કાર્ય કરી શકે, માટે કમ્પાઈલેશનની પ્રક્રિયા દરમિયાન કોઈ ભૂલનો સંદેશ દર્શાવવામાં આવ્યો નહીં. હવે આપણે એ જોઈએ કે જો ખોટો પ્રોગ્રામ ઉમેરવામાં આવે તો શું થાય ? ધારો કે, એડિટરમાં આપણે ઉદાહરણ 10.2ને થોડા ફેરફાર સાથે ઉમેર્યું છે. "circumference = 2 * PI * radius;" વિધાન લખવાને બદલે આપણે ટાઈપિંગ ભૂલ કરીને "circumfernce = 2 * PI * radius;" લખ્યું છે. હવે જો આપણે Ctrl + F7 અથવા F5 કી દબાવીએ તો આકૃતિ 10.19માં દર્શાવ્યા મુજબની સ્ક્રીન રજૂ કરવામાં આવશે.

```

10_2.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 10_2.c

>gcc -pedantic -Os -std=c99 10_2.c -o 10_2
10_2.c: In function 'main':
10_2.c:10:6: error: 'circumfernce' undeclared (first use in this function)
10_2.c:10:6: note: each undeclared identifier is reported only once for each function it appears in
10_2.c:9:11: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
>Exit code: 1

```

આકૃતિ 10.19 : ભૂલ ધરાવતા પ્રોગ્રામનું કમ્પાઈલેશન

સંપૂર્ણ ચિત્ર યોગ્ય રીતે દર્શાવી શકાય તેમ ન હોવાથી અહીં માત્ર આઉટપુટ વિન્ડો જ દર્શાવવામાં આવી છે. માત્ર પરિણામ જોવા માટે Option → Vertical Split વિકલ્પ પસંદ કરી શકાય છે. આકૃતિ 10.19માં ક્ષતિઓ (errors) દર્શાવેલી જોઈ શકાય છે. આકૃતિમાં આપેલી "10_2.c:10:6: error: 'circumfernce' undeclared (first use in this function)" લીટી જુઓ. તે દર્શાવે છે કે circumfernce નામનો ચલ પ્રોગ્રામમાં ઘોષિત કરવામાં આવ્યો નથી. આ ભૂલ સુધારી ફરી પ્રોગ્રામને કમ્પાઈલ કરવાની પ્રક્રિયાનો અમલ કરો. હવે, F5 કી દબાવ્યા પછી કમ્પાઈલરને કોઈ ક્ષતિ નહીં મળે તો તે પ્રોગ્રામનો અમલ કરવાનો પ્રયત્ન કરશે. જો કે, આપણે પ્રોગ્રામમાં scanf() વિધેયનો ઉપયોગ કર્યો હોવાથી આ કિસ્સામાં કોઈ પરિણામ દર્શાવવામાં આવશે નહીં. ઉબુન્ટુની હાલમાં ઉપયોગમાં લેવામાં આવેલી SciTEની આવૃત્તિમાં ક્ષતિ (bug) હોવાનું જણાય છે, કારણ કે તે ઉપયોગકર્તા પાસેથી ઇનપુટ માટેની રાહ જોતું નથી. કોઈ અનુલંબન વગર ફાઈલના નામનો ઉપયોગ કરી આપણે ટર્મિનલ દ્વારા પ્રોગ્રામનો અમલ કરી શકીએ છીએ.

સી ભાષાનો ઇતિહાસ (History of C)

સી પ્રોગ્રામ વિશેનો અભ્યાસ કર્યા બાદ, હવે તેના ઇતિહાસ વિશે સંક્ષિપ્ત માહિતી મેળવીએ. સી ભાષાના મૂળ ઈ. સ. 1972માં બેલ લેબોરેટરી (Bell laboratory)માં નંખાયાં હતાં. સી ભાષાની રચનાનું શ્રેય ડેનિસ એમ. રિચી (Dennis M. Ritchie)ને આપી શકાય. Basic Combined Programming Language (BCPL) નામે ઓળખાતી ભાષા પરથી સી ભાષાની રચના કરવામાં આવી હતી. સી ભાષાની રચનાનો હેતુ એક પુષ્ટ (robust) સિસ્ટમ સોફ્ટવેર બનાવવાનો હતો. પરંતુ પછીના વર્ષોમાં તે પ્રોગ્રામરોની પ્રિય ભાષા બની રહી અને તમામ પ્રકારના સોફ્ટવેર બનાવવા માટે તેનો ઉપયોગ થવા લાગ્યો. માટે જ તેને General Purpose Programming Language તરીકે ઓળખવામાં આવી.

ઈ. સ. 1972માં રચિત આ ભાષાને 1989માં American National Standard Institute (ANSI) ધારાધોરણ દ્વારા પ્રમાણભૂત કરવામાં આવી. ત્યાર પછી તેને આન્સી સી (ANSI C) તરીકે ઓળખવામાં આવી. આજે આ ધારાધોરણને જુદી જુદી ઓપરેટિંગ સિસ્ટમ અને કમ્પાઈલર દ્વારા સમર્થન આપવામાં આવે છે.

સી એક સંરચિત (structured) ભાષા છે. તેમાં પ્રોગ્રામને વિધેય નામે ઓળખાતા નાના વિભાગોમાં વહેંચવાની સુવિધા છે. એકવાર આ વિધેયની રચના કર્યા બાદ તેનો પુનઃ ઉપયોગ શક્ય છે. આવા વિધેયનો સમૂહ સી પ્રોગ્રામની રચના કરે છે. જ્યારે કોઈ ક્ષતિનું નિવારણ કરવાનું હોય ત્યારે આ અભિગમ દ્વારા સમગ્ર પ્રોગ્રામને બદલે માત્ર એક વિધેય ઉપર ધ્યાન કેન્દ્રિત કરી શકાય છે. સામાન્ય રીતે સી ભાષામાં લખવામાં આવેલા કોઈ પણ પ્રોગ્રામને નહિવત્ ફેરફારો સાથે અન્ય ઓપરેટિંગ સિસ્ટમ કે કમ્પાઈલર ધરાવતાં જુદાં જુદાં મશીન પર સરળતાથી ચલાવી શકાય છે. આ ગુણ સી ભાષાને પોર્ટેબલ (portable) બનાવે છે. કેટલાક સી ભાષાને મીડલ લેવલ લેંગ્વેજ (middle level language) માને છે તો કેટલાક માટે તે હાઈર લેવલ લેંગ્વેજ (higher level language) છે. એક પ્રોગ્રામરને જરૂરી એવી તમામ સુવિધાઓ કોઈ પણ પ્રકારે સી ભાષાના ઉપયોગથી ઉપલબ્ધ છે.

સારાંશ

આ પ્રકરણમાં આપણે સી પ્રોગ્રામમાં બંધારણ વિશે અભ્યાસ કર્યો. સી પ્રોગ્રામના વિભાગ સ્વરૂપે ઉપયોગમાં લઈ શકાતા વિવિધ ઘટકો વિશે જાણકારી મેળવી. ત્યાર પછી આપણે સી ભાષાના મૂળાક્ષરો વિશે પણ અભ્યાસ કર્યો, જે અક્ષરો, અંકો, વ્હાઈટ સ્પેસ અને વિશિષ્ટ ચિહ્નો એમ ચાર વિભાગમાં વહેંચાયેલા છે. આપણે આ મૂળાક્ષરોનો ઉપયોગ કરી સી ભાષામાં શબ્દોની રચના કરતા શીખ્યા. ત્યારબાદ આપણે સી પ્રોગ્રામની રચના અને અમલીકરણનાં પગલાં જોયાં. પ્રોગ્રામને કંપાઈલ તથા અમલ કરવા માટે gcc કમ્પાઈલરનો ઉપયોગ કર્યો. અંતમાં આપણે સી પ્રોગ્રામ લખવા અને તેનો અમલ કરવા SciTE ટેક્સ્ટ એડિટરનો ઉપયોગ કરતા શીખ્યા.

શિક્ષકોને સૂચના

SciTE ટેક્સ્ટ એડિટરની ચર્ચા અહીં એમ માનીને કરવામાં આવી છે કે આ એડિટર કમ્પ્યુટરમાં પ્રસ્થાપિત છે અને તેનો શોર્ટકટ ઉપલબ્ધ છે. SciTE એડિટરમાં સી પ્રોગ્રામ લખવાનું શરૂ કરતાં પહેલાં એ નિશ્ચિત કરી લેવું જરૂરી છે કે તેમાં ઓછામાં ઓછું એકવાર Language → C/C++ વિકલ્પની પસંદગી કરેલી હોય. આમ કરતી વખતે સી પ્રોગ્રામ લખતી વખતે કી-વર્ડને પ્રકાશિત (highlight) કરી દર્શાવવામાં આવે છે.

આ ઉપરાંત F5 કે Tools → Go વિકલ્પના ઉપયોગથી SciTE માં જો સીધો જ પ્રોગ્રામનો અમલ કરાવવા ઇચ્છતા હોઈએ તો પ્રોપર્ટીઝ ફાઈલમાં નીચે જણાવેલ લીટી ઉમેરવી જરૂરી બનશે :

command.go.needs.*.c=gcc \$(ccopts) -std=c99 \$(FileNameExt) -o \$(FileName)

નીચે દર્શાવેલ પગલાં દ્વારા પ્રોપર્ટીઝ ફાઈલ બદલી શકાશે :

1. તમારા કમ્પ્યુટરમાં cpp.properties ફાઈલનું સ્થાન શોધી કાઢો. (અમારા કમ્પ્યુટરમાં તે /usr/share/scite/cpp.properties છે.)
2. ટર્મિનલ ખોલી તેમાં sudo edit your_file_path/cpp.properties ટાઈપ કરી એન્ટર કી દબાવો.
3. અહીં એડમિનિસ્ટ્રેટરનો પાસવર્ડ પૂછવામાં આવશે.
4. gedit વિન્ડોમાં cpp.properties ફાઈલ ખુલશે. કોષ્ટક 10.7માં દર્શાવ્યા મુજબનો કોડ ફાઈલમાં શોધી કાઢો.

```
ccopts=-pedantic -Os
cc=g++ $(ccopts) -c $(FileNameExt) -o $(FileName).o
ccc=gcc $(ccopts) -c $(FileNameExt) -o $(FileName).o

make.command=make
command.compile.*.c=$(ccc) -std=c99
command.build.*.c=$(make.command)
command.build.*.h=$(make.command)
#command.go.*.c=/a.out
command.go.*.c=/${FileName}
```

કોષ્ટક 10.7 : cpp.propertiesમાં શોધવાનો કોડ

તમારા કમ્પ્યુટરમાં છેલ્લી બે લીટી આ જ પ્રમાણેની છે તેની ખાતરી કરો. જો તે જુદી હોય તો તેને કોષ્ટક 10.7માં દર્શાવ્યા મુજબ લખો. આ એક એવી વ્યવસ્થા છે કે જેમાં SciTEનો ઉપયોગ કરતી વખતે એક્ઝિક્યુટેબલ (આઉટપુટ) ફાઈલનું નામ સોર્સ ફાઈલના નામનો ઉપયોગ કરીને આપવામાં આવશે.

5. આ ફાઈલમાં નીચેની લીટી ઉમેરી ફાઈલનો સંગ્રહ કરો :

To make the Go command both compile (if needed) and execute, use this setting :

```
command.go.needs.*.c=gcc $(ccopts) -std=c99 $(FileNameExt) -o $(FileName)
```

6. gedit અને ટર્મિનલ વિન્ડો બંધ કરો. SciTE એડિટર હવે તૈયાર છે.

આ પ્રકરણમાં આપવામાં આવેલી સ્ક્રીન એ નમૂનારૂપ સ્ક્રીન છે. શાળામાં ઉપલબ્ધ ઉબુન્ટુની આવૃત્તિ મુજબ તે અલગ હોઈ શકે છે. પરંતુ સ્ક્રીનનું કાર્ય એકસમાન જ રહે છે.

સ્વાધ્યાય

1. સી પ્રોગ્રામની લાક્ષણિકતાની યાદી બનાવી સમજૂતી આપો.
2. main() વિધેયનું મહત્વ સમજાવો.
3. સી પ્રોગ્રામમાં 'ફાઈલ ઇન્કલુડ' વિભાગનો હેતુ શું છે ?
4. આઈડેન્ટિફાયર એટલે શું ? સી પ્રોગ્રામમાં તે કેવી રીતે ઉપયોગી છે ?
5. ચલ એટલે શું ? ચલ વ્યાખ્યાયિત કરવાના નિયમો જણાવો.
6. એક અક્ષર અને સ્ટ્રિંગ અચળ વચ્ચેનો તફાવત જણાવો.

7. નીચેનાં વિધાનો સાચાં છે કે ખોટાં તે જણાવો :

- (a) સી પ્રોગ્રામને ".h" અનુલંબન આપવામાં આવે છે.
- (b) સામાન્ય રીતે સી વિધાનો અલ્પવિરામથી પૂરાં કરવામાં આવે છે.
- (c) Amount એ ચલનું યોગ્ય નામ છે.
- (d) #define PI 3.24 દ્વારા સી પ્રોગ્રામમાં PI નામની ફાઈલ ઉમેરવામાં આવશે.
- (e) "X" એ એક અક્ષરનું યોગ્ય ઉદાહરણ છે.

8. આપેલ વિકલ્પોમાંથી સાચો વિકલ્પ પસંદ કરો :

(1) સી પ્રોગ્રામ ફાઈલને નીચેનામાંથી કયું અનુલંબન આપવામાં આવે છે ?

- (a) c (b) h (c) s (d) t

(2) સી મૂળાક્ષરોને કેટલા વિભાગમાં વર્ગીકૃત કરી શકાય ?

- (a) 0 (b) 2 (c) 4 (d) 8

(3) “=” નિશાની સી મૂળાક્ષરોના કયા વર્ગમાં સમાવિષ્ટ છે ?

- (a) અક્ષરો (b) ખાલી જગ્યા
- (c) વિશિષ્ટ ચિહ્નો (d) અંક

(4) સી ભાષામાં નીચેનામાંથી કયો શબ્દ યોગ્ય કી-વર્ડ છે ?

- (a) ofsize (b) sizeof (c) forsize (d) sizefor

(5) સી ભાષા માટે નીચેનામાંથી કયો ચલ અયોગ્ય છે ?

- (a) Register (b) RegIster (c) registre (d) register

(6) સી ભાષા માટે નીચેનામાંથી કયો પૂર્ણ અચળ અયોગ્ય છે ?

- (a) OXG (b) OxA (c) OxB (d) OxD

(7) સી ભાષા માટે નીચેનામાંથી કયો અપૂર્ણ અચળ યોગ્ય છે ?

- (a) -2.0.5e5 (b) -20.5e5.5 (c) -20.5e5 (d) -20.5e.5

(8) સી ભાષા માટે નીચેનામાંથી કયો અચળ એક અક્ષર રજૂ કરે છે ?

- (a) 'a' (b) ^a' (c) "a" (d) a અને b બંને

(9) સી ભાષામાં નીચેનામાંથી શું વ્યાખ્યાયિત કરવા માટે #define પ્રી-પ્રોસેસર ડાઈરેક્ટિવનો ઉપયોગ કરી શકાય ?

- (a) સ્ટ્રિંગ અચળ (b) સાંકેતિક અચળ
- (c) પૂર્ણ અચળ (d) એક અક્ષર

(10) નીચેનામાંથી કઈ કી દ્વારા પ્રોગ્રામનો સીધો જ અમલ કરી શકાય છે ?

- (a) F7 (b) F9 (c) F5 (d) F8

પ્રાયોગિક સ્વાધ્યાય

1. તમારું નામ, શાળાનું નામ, ધોરણ અને શાળાનું સરનામું સ્ક્રીનની મધ્યમાં દર્શાવવા માટેનો સી પ્રોગ્રામ બનાવો.

* Name : *

* School Name : *

* Standard : *

* Address : *

* *

2. સ્ક્રીન પર તમારી અટકનો પ્રથમ અક્ષર દર્શાવવા માટેનો સી પ્રોગ્રામ બનાવો. ઉદાહરણ તરીકે જો તમારી અટકનો પ્રથમ અક્ષર P હોય તો નીચે મુજબ પરિણામ દર્શાવવું જોઈએ.

* *

* *

*

*

*

3. તમારી પસંદગીનો શુભેચ્છા સંદેશ સ્ક્રીન પર દર્શાવવા માટેનો સી પ્રોગ્રામ લખો. ઉદાહરણ તરીકે, જો તમે કોઈને દિવાળી પર શુભેચ્છા પાઠવવા ઇચ્છો છો તો "Wishing you a Happy and Prosperous Diwali" સંદેશ દર્શાવો.





સી ભાષામાં ડેટા પ્રકાર, પ્રક્રિયકો અને પદાવલિઓ

આ પહેલાના પ્રકરણમાં આપણે કેટલાક સરળ સી પ્રોગ્રામ જોયા અને સી ભાષાના મૂળાક્ષરો તથા ટોકન વિશે પણ અભ્યાસ કર્યો. તેમાંનાં એક ટોકન 'આઈડેન્ટિફાયર'(Identifier)ની પણ આપણે ચર્ચા કરેલી. તમામ સી પ્રોગ્રામમાં આઈડેન્ટિફાયરનો ઉપયોગ કરવામાં આવે છે. ધારો કે, પ્રોગ્રામમાં આપણે `date` નામનો આઈડેન્ટિફાયર વ્યાખ્યાયિત કરવો છે, જે તારીખની કિંમતનો સંગ્રહ કરવા સક્ષમ હોય. આ આઈડેન્ટિફાયરમાં કઈ કિંમતનો સંગ્રહ થવો જોઈએ ? આપણે તેમાં 12.50 કિંમત ઉમેરી શકીએ ? તો જવાબ છે, 'ના'. આ માટે યોગ્ય કિંમત 12 કે 13 જેવી 1થી 31 વચ્ચેની કોઈ પણ પૂર્ણાંક સંખ્યા હશે. આ ઉદાહરણ દર્શાવે છે કે, માત્ર આઈડેન્ટિફાયર નહીં, તેમાં સંગ્રહવામાં આવેલી કિંમત પણ મહત્વની છે. સી ભાષા કેટલાક કી-વર્ડનો ઉપયોગ કરી આઈડેન્ટિફાયરમાં સંગૃહીત કિંમતનો પ્રકાર નિશ્ચિત કરે છે. આ કી-વર્ડ નિશ્ચિત પ્રકારના ડેટા માટે રચવામાં આવેલા હોવાથી તેને ડેટા પ્રકાર (Data type) કહે છે. સી ભાષામાં ઉપલબ્ધ જુદાં જુદાં ડેટા પ્રકારો વિશે આપણે આ પ્રકરણમાં ચર્ચા કરીશું.

ડેટા પ્રકાર શું છે ? (What is Data Type ?)

આઈડેન્ટિફાયરમાં સંગ્રહ કરી શકાય તે કિંમતના પ્રકારને ડેટા પ્રકાર (Data type) તરીકે ઓળખવામાં આવે છે. જો આઈડેન્ટિફાયર `date`માં 12 સંખ્યાનો સંગ્રહ કરવામાં આવે તો તેનો ડેટા પ્રકાર પૂર્ણાંક છે તેમ કહેવાય. આ જ રીતે આઈડેન્ટિફાયર `amount`ને 99.50 કિંમત આપેલ હોય તો તેનો ડેટા પ્રકાર અપૂર્ણાંક છે તેમ કહેવાય.

સી ભાષા વિગતને તેની કિંમત સાથે જોડવા માટે કી-વર્ડનો ઉપયોગ કરે છે. આ કી-વર્ડ બે વસ્તુઓ વ્યાખ્યાયિત કરે છે : આઈડેન્ટિફાયરમાં સંગૃહીત કિંમતનો પ્રકાર અને આઈડેન્ટિફાયર દ્વારા જરૂરી એવી મેમરી જગ્યા. સી ભાષામાં દરેક ડેટા પ્રકારને નિશ્ચિત મેમરી સ્થાન આપવામાં આવે છે. તેનો નિર્દેશ બાઈટ(byte)માં કરવામાં આવે છે. 8 બીટ(Bit)ના સમૂહને 1 બાઈટ કહે છે.

સી ભાષાના મૂળભૂત ડેટા પ્રકાર (Basic Data types of C)

સી ભાષા પૂર્ણાંક (integer), અપૂર્ણાંક (decimal) અને અક્ષર(character)ના નામે ઓળખાતા ડેટા પ્રકારને સમર્થન આપે છે. આ ડેટા પ્રકારને અનુક્રમે `int`, `float` અને `char` કી-વર્ડ દ્વારા રજૂ કરવામાં આવે છે. સી ભાષામાં આ ઉપરાંત `void` નામનો એક અન્ય પ્રાથમિક ડેટા પ્રકાર પણ આપવામાં આવ્યો છે. આઈડેન્ટિફાયરને તેના ડેટા પ્રકાર સાથે સાંકળવા માટે નીચે આપેલ વાક્યરચનાનો ઉપયોગ કરવામાં આવે છે.

Data type identifier 1 [, identifier 2, identifier 3,, identifier n];

અહીં, ચોરસ કોંસમાં આપવામાં આવેલ લખાણ વૈકલ્પિક છે. હવે આપણે તમામ મૂળભૂત ડેટા પ્રકારો વિશે માહિતી મેળવીએ.

પૂર્ણાંક (Integer)

આ પહેલાના મુદ્દામાં આપણે જોયું કે "date" આઈડેન્ટિફાયરને 12 કિંમત આપી શકાય છે. અહીં 12 એ પૂર્ણાંક સંખ્યા છે. ધન કે ઋણ સંપૂર્ણ સંખ્યા કે જેમાં અપૂર્ણાંક ભાગ આપવામાં આવ્યો નથી તેને પૂર્ણાંક કહે છે. -99, 12, -10, 900, 30000 વગેરે પૂર્ણાંક સંખ્યાનાં ઉદાહરણો છે. પૂર્ણાંક પ્રકારના ચલની ઘોષણા માટે નીચે જણાવેલ વાક્યરચનાનો ઉપયોગ કરી શકાય છે :

int identifier 1, [identifier 2, identifier 3,, identifier n];

આ વિધાનને સી ભાષામાં ઘોષણા વિધાન (declaration statement) કહે છે. ઘોષણા વિધાનનાં કેટલાંક ઉદાહરણ નીચે મુજબ છે :

`int roll_number;`

`int date, month, year;`

પ્રથમ વિધાન "roll_number" નામ સાથે એક આઈડેન્ટિફાયરની ઘોષણા કરે છે, જે પૂર્ણાંક સંખ્યાનો સંગ્રહ કરવા સક્ષમ છે. બીજું વિધાન પૂર્ણાંક સંખ્યાનો સંગ્રહ કરી શકે તેવા ત્રણ આઈડેન્ટિફાયર "date", "month" અને "year"ની ઘોષણા કરે છે.

ANSI સી માં int ડેટા પ્રકાર 4 બાઈટ જેટલા મેમરી સ્થાનનો ઉપયોગ કરે છે. તેનો વિસ્તાર -2147483648થી +21474836411 છે. અત્યાર સુધીમાં વ્યાખ્યાયિત કરેલા તમામ ચલ ચિહ્નિત (નિશાનીવાળા -signed) છે એટલે કે તેમાં ધન અને ઋણ બંને પ્રકારની કિંમતોનો સંગ્રહ કરી શકાય છે. ક્યારેક આપણને ઋણ સંખ્યાઓની બિલકુલ જરૂર ન હોય એમ પણ બને. અહીં આપણે ઘોષિત કરેલા આ એક પણ ચલમાં ઋણ સંખ્યાનો સંગ્રહ કરી શકાય તેમ નથી. સી ભાષામાં ચલ સાથે માત્ર ધન સંખ્યાઓને જ સાંકળી શકાય તે પણ શક્ય છે. આ માટે unsigned કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. ઉપરના ઉદાહરણને હવે નીચે દર્શાવ્યા મુજબ સુધારી શકાય :

```
unsigned int roll_number;
```

```
unsigned int date, month, year;
```

આ વિધાનો દર્શાવે છે કે 'roll_number', 'date', 'month' અને 'year' પૂર્ણાંક પ્રકારના આઈડેન્ટિફાયર છે, જે માત્ર ધન સંખ્યાઓનો સંગ્રહ કરવાની ક્ષમતા ધરાવે છે. પૂર્ણાંક ડેટા પ્રકારની સંક્ષિપ્ત માહિતી કોષ્ટક 11.1માં આપવામાં આવી છે.

Data Type	Range	Bytes required	Example
int	-2147483648 to +2147483647	4	int balance_amount;
unsigned int	0 to 4294967295	4	unsigned int counter;

કોષ્ટક 11.1 : પૂર્ણાંક ડેટા પ્રકાર

કોષ્ટક 11.1માં આપેલ કિંમતનો વિસ્તાર ઉપયોગમાં લેવામાં આવેલા બાઈટની સંખ્યા પર અવલંબે છે. ચિહ્નિત (signed) પૂર્ણાંક માટે કિંમતોનો વિસ્તાર $-2^{(n-1)}$ to $+2^{(n-1)} - 1$ દ્વારા ગણી શકાય છે. અહીં "n" એ જરૂરી બીટની સંખ્યાનો નિર્દેશ કરે છે. આ જ રીતે અચિહ્નિત (unsigned) પૂર્ણાંક માટે વિસ્તારની ગણતરી 0થી $2^n - 1$ દ્વારા કરી શકાય છે. પૂર્ણાંક ડેટા પ્રકારની આગળ long કી-વર્ડ ઉમેરવાથી પૂર્ણાંક સંખ્યાઓનો વિસ્તાર વધારી શકાય છે. intની આગળ long ઉમેરવાથી ચલ માટે 8 બાઈટ જેટલું મેમરી સ્થાન આરક્ષિત કરવામાં આવે છે. આ પ્રકારના ચલને નીચે જણાવેલ વાક્યરચના દ્વારા વ્યાખ્યાયિત કરી શકાય :

```
long int population;
```

પૂર્ણાંક ડેટા પ્રકારના જુદા જુદા પ્રકારનાં સ્વરૂપો જોયા પછી હવે એક નમૂનારૂપ પ્રોગ્રામનો અભ્યાસ કરીએ. પૂર્ણાંક ડેટા પ્રકારના ઉપયોગ દર્શાવતા પ્રોગ્રામનું કોડ લિસ્ટિંગ આકૃતિ 11.1માં આપેલ છે.


```
11_1.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_1.c
/* Example 1: Program to illustrate use of integer data type */

#include <stdio.h>
int main( )
- {
    int mark = -10;
    unsigned int date = 30;
    long int population = 42949672950;
    printf(" Mark = %d", mark);
    printf("\n Date = %u", date);
    printf("\n Population = %ld\n\n", population);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.1 : પૂર્ણાંક ડેટા પ્રકારનો ઉપયોગ દર્શાવતો પ્રોગ્રામ

સમજૂતી (Explanation)

આકૃતિ 11.1માં આવેલા દરેક વિધાનની સમજૂતી મેળવીએ. ઉઘડતા છગડિયા કૌંસ પછીનું પ્રથમ વિધાન "mark" નામના પૂર્ણાંક ચલને ઘોષિત કરે છે. આ જ વિધાનમાં ચલની કિંમત -10 પણ આપી છે. બીજું વિધાન "date" નામના પૂર્ણાંક ચલની ઘોષણા કરી તેમાં કિંમત 30 આપે છે. ત્રીજું વિધાન "population" નામના ચલને ઉચ્ચ વિસ્તાર સાથે ઘોષિત કરી તેમાં 42949672950 કિંમતનો સંગ્રહ કરે છે. પછીના ત્રણ વિધાનો દ્વારા આ ત્રણ ચલની કિંમતોને સ્ક્રીન પર દર્શાવવામાં આવે છે. ઉદાહરણ 11.1નું પરિણામ આકૃતિ 11.2માં દર્શાવ્યું છે.

```
11_1.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_1.c
>gcc -pedantic -Os -std=c99 11_1.c -o 11_1
>Exit code: 0
>./11_1
Mark = -10
Date = 30
Population = 42949672950
>Exit code: 0
```

આકૃતિ 11.2 : ઉદાહરણ 11.1નું પરિણામ

અપૂર્ણાંક (Real)

આપણને કેટલીકવાર પૂર્ણાંકને બદલે અપૂર્ણાંક સંખ્યાઓનો ઉપયોગ કરવાની જરૂર પડે છે. ઉદાહરણ તરીકે, કોઈ વ્યક્તિએ ચુકવવાની રકમ 95 કે 95.50 હોઈ શકે. આ પરિસ્થિતિમાં int ડેટા પ્રકાર કાર્ય કરી શકતો નથી. સી ભાષામાં અપૂર્ણાંક સંખ્યાઓનો ઉપયોગ કરવા માટે float કી-વર્ડ દ્વારા વ્યાખ્યાયિત કરવામાં આવતો ડેટા પ્રકાર આપવામાં આવ્યો છે. તે 4 બાઈટ જેટલા સંગ્રાહક સ્થાનનો ઉપયોગ કરે છે. અપૂર્ણાંક સંખ્યાઓ દશાંશચિહ્ન પછી 6 અને દશાંશચિહ્ન પહેલાં 7 અંકો જેટલી ચોકસાઈ ધરાવે છે. જો એથી વધુ ચોકસાઈની જરૂર હોય તો floatના સ્થાને double કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. તે float ડેટા પ્રકારનું વિસ્તરણ છે. double કિંમતો 8 બાઈટનો ઉપયોગ કરે છે અને દશાંશ પછી 16 તથા દશાંશ પહેલાં 17 અંકોની ચોકસાઈ ધરાવે છે. અપૂર્ણાંક ચલનાં ઉદાહરણ નીચે મુજબ છે :

float amount_to_pay;

double balance_amount;

પ્રથમ વિધાન અપૂર્ણાંક સંખ્યાનો સંગ્રહ કરવા માટે સક્ષમ એવા "amount_to_pay" નામના ચલને ઘોષિત કરે છે. બીજું વિધાન "balance_amount" નામના ચલને ઘોષિત કરે છે જે વધુ વિસ્તાર અને ચોકસાઈ સાથે અપૂર્ણાંક સંખ્યાનો સંગ્રહ કરવા સક્ષમ છે.

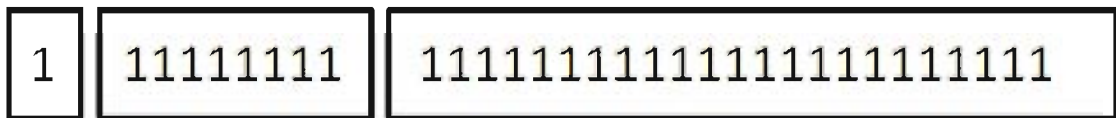
double કી-વર્ડની આગળ long શબ્દ પણ ઉમેરી શકાય છે. આમ કરવાથી doubleનો વિસ્તાર વધારી શકાય છે. float ડેટા પ્રકારની સંક્ષિપ્ત સમજૂતી કોષ્ટક 11.2માં આપવામાં આવી છે.

Data Type	Range	Significant Digits	Bytes required	Example
float	+/-3.4e-38 to +/-3.4e+38	6	4	float price;
double	+/-1.7e-308 to +/-1.7e+308	16	8	double tot_price;
long double	+/-3.4e-4932 to +/-1.1e+4932	16	16	long double credit;

કોષ્ટક 11.2 : float ડેટા પ્રકાર

કોષ્ટક 11.2માં વિસ્તારને વૈજ્ઞાનિક સ્વરૂપમાં દર્શાવ્યો છે. સી ભાષા વૈજ્ઞાનિક સ્વરૂપે રહેલી સંખ્યાને પણ સમર્થન આપે છે. ઉદાહરણ તરીકે 95.50 સંખ્યાને 0.9550e2 સ્વરૂપમાં પણ રજૂ કરી શકાય છે. અહીં 0.9550ને મેન્ટિસા (mantissa) તરીકે તથા 2ને એક્સ્પોનન્ટ (exponent) તરીકે ઓળખવામાં આવે છે. અપૂર્ણાંક સંખ્યા ધન કે ઋણ હોઈ શકે છે. આકૃતિ 11.4માં અપૂર્ણાંક સંખ્યાની મેમરી રજૂઆત દર્શાવી છે.

31 30 23 22 0



Sign Exponent

Mantissa

આકૃતિ 11.3 : અપૂર્ણાંક સંખ્યાની મેમરી રજૂઆત

હવે, float, double અને long double ડેટા પ્રકારનો ઉપયોગ દર્શાવતું એક ઉદાહરણ જોઈએ. આકૃતિ 11.4માં float ડેટા પ્રકારને સ્પષ્ટ કરતા પ્રોગ્રામનું કોડ લિસ્ટિંગ દર્શાવ્યું છે.

```
11_2.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_2.c
/* Example 2: Program to illustrate use of real data type */

#include <stdio.h>
int main( )
- {
    /* First three statements declares and initializes of variables first, second and third */

    float first = 9876543210987654321.987654321;
    double second = 9876543210987654321.987654321;
    long double third = 9876543210987654321.987654321;

    printf(" First = %f", first);
    printf("\n Second = %lf", second);
    printf("\n Third = %Lf\n", third);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.4 : float ડેટા પ્રકારને સ્પષ્ટ કરતો પ્રોગ્રામ

સમજૂતી (Explanation)

ઉદાહરણ છગડિયા કોંસ પછીનાં ત્રણ વિધાનો "first", "second" અને "third" નામના ત્રણ ચલને ઘોષિત કરે છે. આ દરેક ચલમાં અપૂર્ણ સંખ્યા 9876543210987654321.987654321નો સંગ્રહ કરવામાં આવ્યો છે. પછીનાં ત્રણ વિધાનો આ ચલની કિંમતોને સ્ક્રીન પર દર્શાવે છે. આકૃતિ 11.5માં ઉદાહરણ 11.2નું પરિણામ દર્શાવ્યું છે.

```
11_2.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_2.c
>gcc -pedantic -Os -std=c99 11_2.c -o 11_2
>Exit code: 0
>./11_2
First = 9876543516404875264.000000
Second = 9876543210987655168.000000
Third = 9876543210987655168.000000
>Exit code: 0
```

આકૃતિ 11.5 : ઉદાહરણ 11.2નું પરિણામ

અહીં જોઈ શકાય છે કે, ચલ 'first', 'second' અને 'third'ની કિંમતો અનુક્રમે 9876543516404875264.000000, 9876543210987655168.000000 અને 9876543210987655168.000000 દર્શાવવામાં આવી છે. પરિણામને ધ્યાનથી જોતાં જાણી શકાય છે કે "first" ચલની મૂળ કિંમત સાથે પ્રથમ માત્ર 7 અંકોની સમાનતા છે. એ જ પ્રમાણે "second" અને "third" ચલના પ્રથમ માત્ર 15 અંકો જ સમાન છે. આમ થવાનું કારણ એ છે કે float ડેટા પ્રકાર દશાંશ પછી 6 અને દશાંશ પહેલાં 7 અંકોની ચોકસાઈ ધરાવે છે. આ જ રીતે double અને long float ડેટા પ્રકાર દશાંશ પછી 15 અને દશાંશ પહેલાં 16 અંકોની ચોકસાઈ ધરાવે છે.

અપૂર્ણાંક સંખ્યાઓનો ઉપયોગ કેટલીકવાર ચોકસાઈને હાનિ પહોંચાડે છે. તેથી આપણે અપેક્ષિત નિશ્ચિત પરિણામ મેળવી શકતા નથી. માટે જ પરિણામની વધુ ચોકસાઈ માટે ઉચ્ચ પરિશુદ્ધતા (precision) ધરાવતા ડેટા પ્રકારનો ઉપયોગ હિતાવહ છે. અપૂર્ણાંક અંકોનો ઉપયોગ કરવાનો લાભ એ છે કે તે પૂર્ણાંક અંકોની સરખામણીમાં કિંમતોનો વ્યાપક વિસ્તાર રજૂ કરે છે.

અક્ષર (character)

ઉપરના બંને કિસ્સાઓમાં આપણે આંકડાકીય વિગતોનો સંગ્રહ કર્યો. પરંતુ ધારો કે આપણે પુરુષ માટે 'M' (male) અને સ્ત્રી માટે 'F' (female) જેવી જાતિવાચક સંજ્ઞા અથવા તો શહેરનાં નામ જેવી વિગતોનો સંગ્રહ કરવાની જરૂર હોય તો? અહીં આપણી પાસે માત્ર મૂળાક્ષરો છે. આ પ્રકારની કિંમતોનો સંગ્રહ int કે float દ્વારા કરી શકાતો નથી. આ પ્રકારની કિંમતોનો સંગ્રહ કરવા માટે char કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. તે 1 બાઈટ જેટલા મેમરી સ્થાનનો ઉપયોગ કરે છે. દરેક અક્ષર ASCII (American Standard Code for Information Interchange) નામે ઓળખતાં પૂર્ણાંક સાથે સંકળાયેલ હોય છે. ASCII કિંમતોની વિગતો પરિશિષ્ટ-II માં આપવામાં આવી છે. પૂર્વ નિર્ધારિત રીતે char અચિહ્નિત(Unsigned) છે. ચિહ્નિત (signed) char પણ શક્ય છે. કોષ્ટક 11.3માં char ડેટા પ્રકાર વિશે ટૂંકી સમજૂતી આપવામાં આવી છે.

Data Type	Range	Bytes required	Example
char	-128 to + 127	1	char gender;
unsigned char	0 to 255	1	unsigned char choice;

કોષ્ટક 11.3 : char ડેટા પ્રકાર

char ડેટા પ્રકારને સ્પષ્ટ કરતા પ્રોગ્રામનું કોડ ચિસ્ટિંગ આકૃતિ 11.6માં દર્શાવ્યું છે.

```

11_3.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_3.c
/*Example 3: Program to illustrate use of character data type */

#include <stdio.h>
int main( )
- {
    char gender = 'm';
    char answer = 'M';
    char number = '2';
    char sp_char = ' ';
    printf("\nThe character stored in gender is %c and its ASCII value is %d", gender, gender);
    printf("\nThe character stored in answer is %c and its ASCII value is %d", answer, answer);
    printf("\nThe character stored in number is %c and its ASCII value is %d", number, number);
    printf("\nThe character stored in sp_char is %c and its ASCII value is %d\n\n", sp_char, sp_char);
    return 0;
}
/* End of Program */

```

આકૃતિ 11.6 : char ડેટા પ્રકાર દર્શાવતો પ્રોગ્રામ

સમજૂતી (Explanation)

ઉપરોક્ત છગડિયા કૌંસ પછીનાં પ્રથમ ચાર વિધાનો અનુક્રમે "gender", "answer", "number" અને "sp_char" નામના ચાર ચલ ઘોષિત કરી તેને અનુક્રમે 'm', 'M', '2' અને ' ' (ખાલી જગ્યા) આપણે. પછીનાં ચાર વિધાનો આ ચલમાં આવેલી કિંમતોને તેની ASCII કિંમત સાથે સ્ક્રીન પર દર્શાવશે. ઉદાહરણ 11.3નું પરિણામ આકૃતિ 11.7માં દર્શાવ્યું છે.


```
11_3.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_3.c
>gcc -pedantic -Os -std=c99 11_3.c -o 11_3
>Exit code: 0
>./11_3

The character stored in gender is m and its ASCII value is 109
The character stored in answer is M and its ASCII value is 77
The character stored in number is 2 and its ASCII value is 50
The character stored in sp_char is  and its ASCII value is 32

>Exit code: 0
```

આકૃતિ 11.7 : ઉદાહરણ 11.3નું પરિણામ

સ્પેસ m અને કેપિટલ Mની ASCII કિંમતો વચ્ચેના તફાવતની નોંધ કરો. આપણે જોઈ શકીએ છીએ કે અંકોનો ઉપયોગ અક્ષરોના સ્વરૂપે પણ કરી શકાય છે. સંખ્યા 2નો અક્ષર તરીકે ઉપયોગ કરવા માટે તેને '2' સ્વરૂપે લખવામાં આવે છે. અહીં એ પણ જુઓ કે sp_char ચલમાં આવેલ કિંમત દૃશ્યમાન ન હોવા છતાં તેનો આસ્કી અંક દૃશ્યમાન છે.

ખાલી વિગતોનો ગણ (Empty data set)

સી ભાષામાં void કી-વર્ડ દ્વારા એક વિશિષ્ટ ડેટા પ્રકાર પૂરો પાડવામાં આવે છે. આ ડેટા પ્રકાર કોઈ કિંમતનો સંગ્રહ કરતો નથી માટે તેનો 'ખાલી ગણ' તરીકે નિર્દેશ કરવામાં આવે છે. તેનો ઉપયોગ મુખ્યત્વે વિધેયની પરત કિંમત દર્શાવવા માટે કરવામાં આવે છે. આગળના પ્રકરણમાં જણાવ્યું તે મુજબ સી પ્રોગ્રામ વિધેયોનો સમૂહ છે. સી પ્રોગ્રામના વિધેય કોઈ કિંમત પરત કરી શકે છે. જો આપણે ઈચ્છતા હોઈએ કે વિધેય કોઈ કિંમત પરત ન કરે તો તેના નામની પહેલાં void કી-વર્ડ ઉમેરવામાં આવે છે.

આગળનાં તમામ ઉદાહરણોમાં પ્રોગ્રામનું અંતિમ વિધાન return 0; છે. આ વિધાન કંપાઈલરને પ્રોગ્રામમાંથી બહાર નીકળવાનો નિર્દેશ કરે છે. જો આ વિધાન ન લખવામાં આવે તો ચેતવણીનો સંદેશ "Function should return a value" રજૂ કરવામાં આવે છે. આ સંદેશ ન આવે તે માટે main()ની આગળ void ઉમેરી શકાય છે.

ચલને કિંમત આપવી (Assigning values to variable)

પ્રોગ્રામના અમલીકરણ દરમિયાન એક વાર ઘોષિત કર્યા બાદ ચલમાં કિંમતો આપી શકાય છે. આ કિંમતો ચલના ડેટા પ્રકારને અનુરૂપ હોવી જોઈએ. નીચેની વાક્યરચના દ્વારા ચલમાં કિંમત આપી શકાય છે :

Variable = Value;

એક જ વિધાનમાં ચલની ઘોષણા કરી તેમાં કિંમત આપવી પણ શક્ય છે. આમ કરવા માટે નીચેની વાક્યરચનાનો ઉપયોગ કરી શકાય :

data type variable = value;

અત્યાર સુધી આ પ્રકરણમાં ચર્ચવામાં આવેલાં તમામ ઉદાહરણોમાં ચલનો પ્રારંભ આ તકનિકથી કરવામાં આવ્યો છે.

ઉપયોગકર્તા દ્વારા નિર્મિત ડેટા પ્રકાર (User defined data type)

સી ભાષા પરિવર્તનક્ષમ (flexible) છે. તે ઉપયોગકર્તાને હયાત ઘટકોમાંથી નવા ઘટકો બનાવવાની સુવિધા આપે છે તથા મૂળભૂત ડેટા પ્રકારનો ઉપયોગ કરી નવા ડેટા પ્રકારની રચના કરી શકાય છે. સી ભાષામાં ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત ડેટા પ્રકારની રચના કરવા માટે typedef અને enum નામના બે કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. બંને પ્રકારના કી-વર્ડ દ્વારા ડેટા પ્રકાર તરીકે ઉપયોગમાં લઈ શકાય તેવા ચલ વ્યાખ્યાયિત કરી શકાય છે.

ટાઈપ ડેફિનેશન (Type definition)

ટાઈપ ડેફિનેશન દ્વારા ડેટા પ્રકાર વ્યાખ્યાયિત કરવા માટેની વાક્યરચના નીચે પ્રમાણે છે :

```
typedef datatype variable;
```

અહીં, datatype એ int કે float જેવા ઉપલબ્ધ ડેટા પ્રકારનો નિર્દેશ કરે છે. ઉપલબ્ધ ડેટા પ્રકારને આપવામાં આવનાર નવા નામનો નિર્દેશ variable દ્વારા કરવામાં આવે છે. ખરી રીતે જોતાં, સી ભાષા નવો ડેટા પ્રકાર રચવાની અનુમતિ આપતું નથી. તેને બદલે તે ઉપલબ્ધ ડેટા પ્રકારને ઉપનામ (alias) આપવાની સુવિધા ધરાવે છે. આપણે સામાન્ય રીતે જીવનમાં પણ ઉપનામોનો ઉપયોગ કરીએ છીએ. વ્યક્તિને આપવામાં આવતું હુલામણું નામ તેનું ઉદાહરણ છે. typedefના કેટલાંક ઉદાહરણ નીચે આપેલાં છે :

```
typedef char alpha;
```

```
typedef double twice;
```

પ્રથમ વિધાન char ડેટા પ્રકારને "alpha" ઉપનામ અપાશે જ્યારે બીજા વિધાન દ્વારા double ડેટા પ્રકારને "twice" ઉપનામ આપવામાં આવશે. આ ઉપનામ આપ્યા બાદ, નીચેનાં વિધાનો દ્વારા અક્ષર માટેનો "choice" નામનો ચલ તથા "amount" નામનો "ડબલ" પ્રકારનો ચલ ઘોષિત કરી શકાય.

```
alpha choice;
```

```
twice amount;
```

ઈન્યુમેરેટેડ ડેટા પ્રકાર (Enumerated data type)

સી ભાષામાં enum કી-વર્ડ દ્વારા ઈન્યુમેરેટેડ ડેટા પ્રકારની રચના કરી શકાય છે. enumનો ઉપયોગ કરવા માટેની વાક્યરચના નીચે મુજબ છે :

```
enum identifier {value 1, value 2, value 3, ....., value n};
```

અહીં enum કી-વર્ડ છે. Identifier એ નવો ડેટા ટાઈપ વ્યાખ્યાયિત કરવા માટે ઉપયોગમાં લેવામાં આવનાર નામ છે. Value 1થી Value n એ આંકડાકીય અચલો 0, 1, 2, વગેરેને આપવામાં આવેલા નામ છે. enum દ્વારા વ્યાખ્યાયિત કરવામાં આવેલા ચલમાં છગડિયા કૌંસમાં આપવામાં આવેલી કોઈ પણ એક કે વધુ કિંમત ઉમેરી શકાય છે. enum ના ઉપયોગનું ઉદાહરણ નીચે મુજબ છે :

```
enum money {rupee, dollar, pound, yen};
```

```
enum money currency;
```

```
currency = dollar;
```

પ્રથમ વિધાન ઈન્યુમેરેટેડ ડેટા પ્રકાર "money" વ્યાખ્યાયિત કરે છે. બીજા વિધાન દ્વારા money પ્રકારનો ઉપયોગ કરી "currency" નામનો ચલ ઘોષિત કરવામાં આવ્યો છે. છેલ્લા વિધાનમાં currency ચલને "dollar" કિંમત આપવામાં આવી છે જે "1"ને સમકક્ષ છે. અહીં, rupee, dollar, pound અને yen અનુક્રમે 0, 1, 2 અને 3 આંકડાકીય અચલો રજૂ કરે છે. આ આંકડાકીય અચલો કંપાઈલર દ્વારા આપોઆપ ઉમેરવામાં આવે છે. નવી કિંમતો ઉમેરી આ અચલોને બદલી પણ શકાય છે. ઉદાહરણ તરીકે જો આપણે rupeeને 10, dollarને 50, poundને 75 અને yenને 100 કિંમત આપવી હોય તો પ્રથમ વિધાનને enum money {rupee = 10, dollar = 50, pound = 75, yen = 100}; સ્વરૂપે લખી શકાય. હવે currency = dollar વિધાન currency = 50ને સમકક્ષ બનશે.

ઉપયોગકર્તા દ્વારા નિર્મિત ડેટા પ્રકારોના ઉપયોગથી સી પ્રોગ્રામની વાચનક્ષમતામાં વૃદ્ધિ કરી શકાય છે. તે ડેટા પ્રકારને અર્થપૂર્ણ નામ સાથે વ્યાખ્યાયિત કરવાની સુવિધા પણ આપે છે.

તારવેલા ડેટા પ્રકાર (Derived data type)

આપણે અગાઉ જોઈ ગયા તેમ સી વિસ્તારી શકાય તેવી (Extensible) ભાષા છે. આપણી ઘણી જરૂરિયાતો માટે

મૂળભૂત ડેટા પ્રકારો પૂરતા બની રહે તેમ બને, પરંતુ જટિલતાને કારણે દરેક સમયે તેનો ઉપયોગ કરવો યોગ્ય નથી. ઉદાહરણ તરીકે, ધારો કે આપણે 15 વિદ્યાર્થીઓના વર્ગ(grades)નો સંગ્રહ કરવા ઇચ્છીએ છીએ. આ માટે આપણે 15 ચલ ઘોષિત કરવાની જરૂર પડે. 15 ચલની સારસંભાળ પ્રોગ્રામની જટિલતામાં વધારો કરે. આ જ રીતે જો આપણે 15 વિદ્યાર્થીઓ માટે તેમના અનુક્રમ, જાતિ, નામ અને સરનામા જેવી સંપૂર્ણ વિગતનો સંગ્રહ કરવો હોય તો વધારાના ચલની જરૂર પડે. અહીં ચલના જૂથ કદાચ મુશ્કેલી ઊભી કરે. તેના નિવારણ માટે સી ભાષા ડેટા પ્રકાર તારવવાની સુવિધા પૂરી પાડે છે. એરે (array), સ્ટ્રક્ચર (structure), યુનિયન (union) અને પોઈન્ટર (pointer) તારવેલાં ડેટા પ્રકારનાં ઉદાહરણ છે.

એરે (Array)

સી ભાષામાં એરે નામના ડેટા બંધારણની રચના શક્ય છે. આ ડેટા બંધારણમાં એકસમાન લાક્ષણિકતા ધરાવતા ચલના સમૂહનો સમાવેશ કરી શકાય છે. એરે વ્યાખ્યાયિત કરવા માટેની વાક્યરચના નીચે મુજબ છે :

data type variable[size];

અહીં datatype એ ડેટાનો કોઈ પણ મૂળભૂત પ્રકાર હોઈ શકે છે. Variable એ એરેનું નામ છે અને size એ એરેમાં આવેલા ઘટકોની કુલ સંખ્યા છે. ઉદાહરણ તરીકે, **char name_of_subject[10];** સી વિધાન name_of_subject નામનો એક 10 કદ ધરાવતો એરે વ્યાખ્યાયિત કરશે. આનો અર્થ એ થાય કે, "name_of_subject" નામનો ચલ વ્યાખ્યાયિત કરી તેમાં 10 અક્ષરોનો સમૂહ સંગ્રહ કરી શકાશે. અહીં એરેનો દરેક ઘટક અક્ષર પ્રકારનો રહેશે.

સી ભાષામાં **char name_of_subject[] = "C Language";** પ્રકારનું વિધાન પણ હોઈ શકે છે. અહીં એરેની લંબાઈ કંપાઈલર દ્વારા આપોઆપ નિશ્ચિત કરી લેવામાં આવશે. આપેલ કિસ્સામાં એરેની લંબાઈ 11 હશે. પ્રકરણ 15માં એરે વિશે વિસ્તૃત ચર્ચા કરવામાં આવી છે.

સી ભાષામાં સ્ટ્રક્ચર, યુનિયન અને પોઈન્ટર જેવા અન્ય તારવેલા ડેટા પ્રકાર પણ ઉપલબ્ધ છે. આ તમામ ડેટા પ્રકારોની ચર્ચા આ પુસ્તકની મર્યાદા બહાર છે.

પ્રક્રિયક અને પદાવલિ (Operators and Expression)

અત્યાર સુધી આપણે સી ભાષામાં ઉપલબ્ધ ડેટા પ્રકારો વિશે જાણકારી મેળવી. હવે આપણે સી ભાષામાં ઉપલબ્ધ જુદા જુદા પ્રક્રિયકો વિશે અભ્યાસ કરીએ. પ્રક્રિયકો અને ઓપરેન્ડ (operand)ના ઉપયોગ દ્વારા પદાવલિ કેવી રીતે બનાવી શકાય તેનો પણ આપણે અભ્યાસ કરીશું. અંતમાં આપણે જોઈશું કે પદાવલિઓનું મૂલ્યાંકન કેવી રીતે કરવામાં આવે છે.

પ્રક્રિયકો (Operators)

શાળામાં પ્રાથમિક ગણિતનો અભ્યાસ કરતી વખતે આપણે બે સંખ્યાઓના સરવાળા કે બાદબાકી જેવી પ્રક્રિયાઓ કરેલી, જેમકે, $5 + 3$ અને $9 - 7$. અહીં 5, 3, 9 અને 7 અચળો છે જ્યારે '+' અને '-' નિશાનીઓ આ અચળો પરની પ્રક્રિયા વ્યાખ્યાયિત કરે છે.

ઓપરેન્ડ પર કરવામાં આવનાર પ્રક્રિયાને વ્યાખ્યાયિત કરતી નિશાનીને પ્રક્રિયક (operators) તરીકે ઓળખવામાં આવે છે. સી ભાષામાં પ્રક્રિયકોનું વર્ગીકરણ નીચે આપેલ આઠ વર્ગોમાં કરી શકાય :

1. ગાણિતિક પ્રક્રિયકો (Arithmetic Operators)
2. નિરૂપક પ્રક્રિયકો (Assignment Operators)
3. સંબંધસૂચક પ્રક્રિયકો (Relational Operators)
4. વધારા / ઘટાડા સૂચક પ્રક્રિયકો (Increment and Decrement Operators)
5. શરતી પ્રક્રિયકો (Conditional Operator)
6. તાર્કિક પ્રક્રિયકો (Logical Operators)
7. બિટવાઈઝ પ્રક્રિયકો (Bitwise Operators)
8. વિશિષ્ટ પ્રક્રિયકો (Special Operators)

આ તમામ પ્રક્રિયકો વિશે ચર્ચા કરીએ.

ગાણિતિક પ્રક્રિયકો (Arithmetic Operators)

આંકડાકીય ગણતરીઓ કરવા માટે સી ભાષામાં '+', '-', '*', '/' અને '%' પ્રક્રિયકો આપવામાં આવ્યા છે. ઘાતીક માટેનો કોઈ પ્રક્રિયક સી ભાષામાં ઉપલબ્ધ નથી. કોષ્ટક 11.4માં આ પ્રક્રિયકો તેમના ઉપયોગ સાથે દર્શાવ્યા છે.

પ્રક્રિયક	ઉપયોગ
+	બે સંખ્યાઓનો સરવાળો અથવા યુનરી ધન
-	બે સંખ્યાઓની બાદબાકી અથવા યુનરી ઋણ
*	બે સંખ્યાઓનો ગુણાકાર
/	બે સંખ્યાઓનો ભાગાકાર કરી ભાગફળ રૂપે પરિણામ
%	બે સંખ્યાઓનો ભાગાકાર કરી શેષ રૂપે પરિણામ

કોષ્ટક 11.4 : આંકડાકીય પ્રક્રિયકો

કોષ્ટક 11.4માં આપવામાં આવેલ પ્રથમ ચાર પ્રક્રિયકો વિશે આપણે માહિતગાર છીએ. સી ભાષા મોડ્યુલો (Modulo) નામે ઓળખાતા એક અતિરિક્ત પ્રક્રિયક '%'ની સુવિધા પૂરી પાડે છે. બે પૂર્ણાંક સંખ્યાઓના ભાગાકારથી મળતી શેષ ગણવા માટે આ પ્રક્રિયકનો ઉપયોગ કરવામાં આવે છે. અહીં એ નોંધ લેવી જરૂરી છે કે આ પ્રક્રિયકનો ઉપયોગ અપૂર્ણ (float) સંખ્યાઓ સાથે શક્ય નથી.

હવે આપણે આ પ્રક્રિયકોનો ઉપયોગ કેવી રીતે કરવો તે જોઈએ. ઉદાહરણ સ્વરૂપે આપેલ નીચેનું સી વિધાન જુઓ :

```
total_cost = quantity * cost_item;
```

આ સી વિધાન સી પદાવલિ(C expression)ની રચના કરે છે. અહીં 'total_cost', 'quantity' અને 'cost_item'ને સી ચલ તરીકે ઓળખવામાં આવે છે. અહીં જોઈ શકાય છે કે પદાવલિમાં બે પ્રક્રિયકો અને ત્રણ ઓપરેન્ડ આવેલાં છે. આ પદાવલિનું મૂલ્યાંકન કરવામાં આવે ત્યારે બે પ્રક્રિયાઓ થાય છે. પ્રથમ, quantity અને cost_item ચલમાં રહેલી કિંમતોનો ગુણાકાર કરવામાં આવે છે અને બીજું, ગુણાકારથી મળેલા પરિણામનો total_cost નામના ચલમાં સંગ્રહ કરવામાં આવે છે.

પદાવલિમાં ઉપયોગમાં લેવામાં આવેલા ઓપરેન્ડને આધારે ગણતરીને ત્રણ વર્ગોમાં વહેંચી શકાય : પૂર્ણાંક ગણતરી (integer arithmetic), અપૂર્ણાંક ગણતરી (real arithmetic) અને મિશ્ર પ્રકારની ગણતરી (mixed mode arithmetic).

પૂર્ણાંક ગણતરી (Integer Arithmetic)

જો પદાવલિમાં ઉપયોગમાં લેવામાં આવેલ ઓપરેન્ડ ધન કે ઋણ પૂર્ણ સંખ્યાઓ હોય તો તેને પૂર્ણાંક ગણતરી (integer arithmetic) તરીકે ઓળખવામાં આવે છે. અહીં, પદાવલિને પૂર્ણાંક પદાવલિ (integer expression) કહે છે. પૂર્ણાંક પદાવલિનું પરિણામ હંમેશાં પૂર્ણાંક મળે છે.

હવે, કુલ કિંમત અને વસ્તુની સંખ્યા આપવામાં આવી હોય ત્યારે વસ્તુની એકમ કિંમત શોધવા માટેનો પ્રોગ્રામ જોઈએ. પૂર્ણાંક ગણતરી રજૂ કરતો આ પ્રોગ્રામ આકૃતિ 11.8માં દર્શાવ્યો છે.

```
11_4.c - ScITE
File Edit Search View Tools Options Language Buffers Help
1 11_4.c
/* Example 4: Program to illustrate use of integer arithmetic */
#include <stdio.h>
int main( )
- {
    /* Use of typedef to create new data type name */
    typedef int whole_number;

    /* Declaration of variables using new data type name */
    whole_number total_cost = 650 , quantity = 24, cost_item;

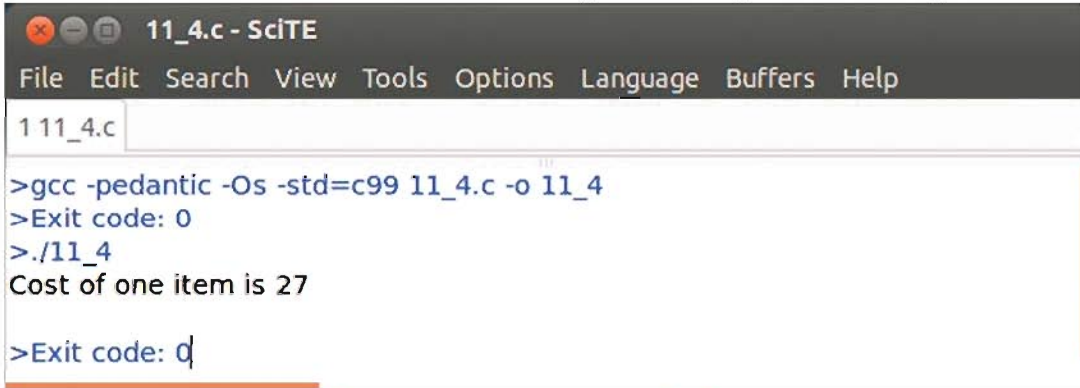
    /* C expression for calculating cost per item */
    cost_item = total_cost / quantity;

    printf("Cost of one item is %d\n", cost_item);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.8 : પૂર્ણાંક ગણતરી દર્શાવતો પ્રોગ્રામ

સમજૂતી (Explanation)

અહીં પ્રથમ વિધાન દ્વારા typedef કી-વર્ડનો ઉપયોગ કરી int માટેનું ઉપનામ આપવામાં આવ્યું છે. ત્યાર પછી નવા ઉપનામનો ઉપયોગ કરી "total_cost", "quantity" અને "cost_item" નામના ત્રણ પૂર્ણાંક ચલને વ્યાખ્યાયિત કરવામાં આવ્યા છે. "total_cost" ચલમાં 650 અને "quantity" ચલમાં 24 કિંમત પણ ઉમેરવામાં આવી છે. પછીનું વિધાન "total_cost" નો "quantity" વડે ભાગાકાર કરી પરિણામનો "cost_item" ચલમાં સંગ્રહ કરે છે. ત્યાર પછી "cost_item" ચલની કિંમતને સંદેશ સાથે દર્શાવવામાં આવી છે. ઉદાહરણ 11.4નું પરિણામ આકૃતિ 11.9માં દર્શાવ્યું છે.



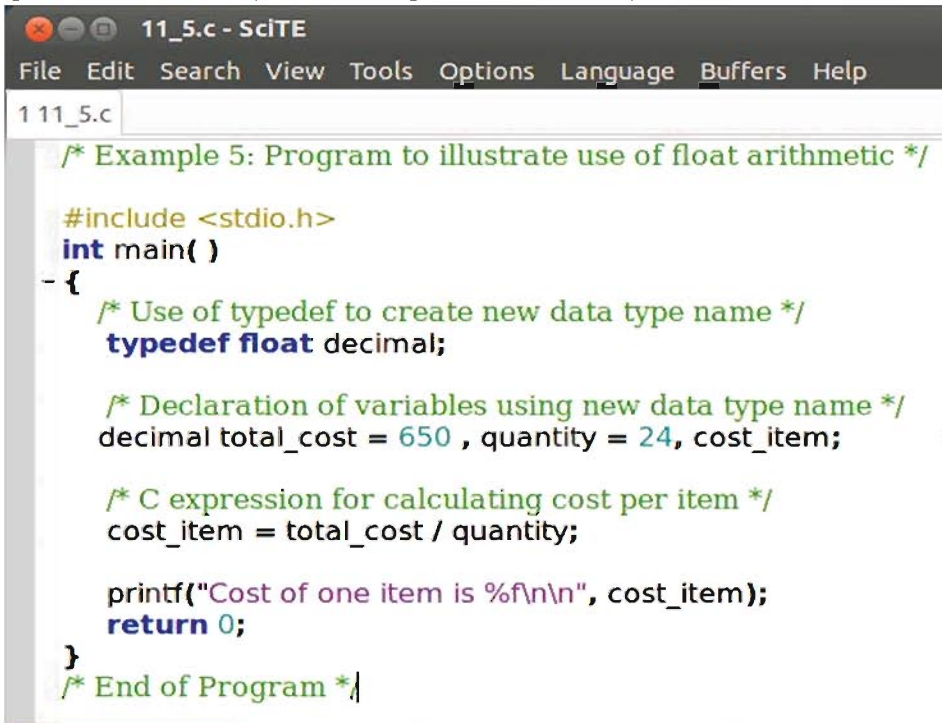
```
11_4.c
>gcc -pedantic -Os -std=c99 11_4.c -o 11_4
>Exit code: 0
>./11_4
Cost of one item is 27
>Exit code: 0
```

આકૃતિ 11.9 : ઉદાહરણ 11.4નું પરિણામ

પારંપરિક ગણિતશાસ્ત્ર મુજબ $650/24$ નું પરિણામ 27.0833333333 મળવું જોઈએ. પરંતુ પરિણામમાં "cost_item" ચલની કિંમત 27 જોઈ શકાય છે. આ ઉપરથી કહી શકાય કે પૂર્ણાંક ગણતરીનું પરિણામ હંમેશા પૂર્ણાંક સંખ્યામાં મળશે. પારંપરિક ગણિત કરતાં સી ભાષામાં ભાગાકાર માટેના પ્રક્રિયકનો ઉપયોગ અલગ છે. પારંપરિક ગણિતમાં $5/10$ નું પરિણામ મળે, જે અહીં 0 મળશે.

અપૂર્ણાંક ગણતરી (Real Arithmetic)

આપેલ પદાવલિમાં ઉપયોગમાં લેવામાં આવેલા પ્રક્રિયકો જો અપૂર્ણાંક હોય તો તેને અપૂર્ણાંક ગણતરી (real arithmetic) કહે છે. અપૂર્ણાંક ગણતરીનું પરિણામ હંમેશા દશાંશચિહ્ન સાથેની કિંમતો દ્વારા દર્શાવવામાં આવે છે. પૂર્ણાંક ગણતરીને અપૂર્ણાંક ગણતરીમાં ફેરવવા માટે આકૃતિ 11.8માં દર્શાવેલ પ્રોગ્રામમાં ફેરફાર કરીએ. સુધારેલો પ્રોગ્રામ આકૃતિ 11.10માં અને તેનું પરિણામ આકૃતિ 11.11માં દર્શાવ્યું છે.



```
11_5.c
/* Example 5: Program to illustrate use of float arithmetic */
#include <stdio.h>
int main( )
{
    /* Use of typedef to create new data type name */
    typedef float decimal;

    /* Declaration of variables using new data type name */
    decimal total_cost = 650 , quantity = 24, cost_item;

    /* C expression for calculating cost per item */
    cost_item = total_cost / quantity;

    printf("Cost of one item is %f\n\n", cost_item);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.10 : અપૂર્ણાંક ગણતરીનું ઉદાહરણ દર્શાવતો પ્રોગ્રામ

```
11_5.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_5.c
>gcc -pedantic -Os -std=c99 11_5.c -o 11_5
>Exit code: 0
>./11_5
Cost of one item is 27.083334
>Exit code: 0
```

આકૃતિ 11.11 : ઉદાહરણ 11.5નું પરિણામ

સમજૂતી (Explanation)

આ ઉદાહરણ એક અપવાદ સિવાય ઉદાહરણ 11.4 જેવું જ છે. અહીં ચલને float સ્વરૂપે ઘોષિત કર્યા છે. "total_cost" અને "quantity" ચલને આપવામાં આવેલી કિંમતો પૂર્ણાંક હોવા છતાં આંતરિક રીતે તેનો સંગ્રહ અપૂર્ણાંક (float) તરીકે થયેલ છે. માટે, પરિણામ પણ અપૂર્ણાંક સંખ્યામાં છે.

મિશ્ર પ્રકારની ગણતરી (Mixed Mode Arithmetic)

આ પ્રકારની ગણતરીમાં પૂર્ણાંક તેમજ અપૂર્ણાંક બંને પ્રકારના ઓપરેન્ડનો ઉપયોગ કરી શકાય છે. અહીં પરિણામનો આધાર ચલને આપવામાં આવેલ કિંમતોના પ્રકાર પર છે. $result = 25.75 * 5$; જેવી પદાવલિ સી ભાષામાં લખવી શક્ય હોવા છતાં તેનું સીધું મૂલ્યાંકન શક્ય નથી. ઉપયોગમાં લેવામાં આવેલાં તમામ ઓપરેન્ડ સમાન ડેટા પ્રકાર ધરાવતા હોય તો જ પદાવલિનું મૂલ્યાંકન શક્ય બને છે. મિશ્ર પ્રકારની ગણતરી કરી શકાતી હોવાથી સી ભાષા આ મુશ્કેલીનું નિવારણ કરે છે. સી ભાષામાં નિમ્ન સ્તરના ડેટા પ્રકારનું ઉચ્ચ ડેટા પ્રકારમાં આપોઆપ રૂપાંતરણ કરવામાં આવે છે.

નિરૂપક પ્રક્રિયકો (Assignment Operators)

અત્યાર સુધી ચર્ચેલાં ઉદાહરણોમાં આપણે $total_cost = quantity * cost_item$; કે $date = 30$; જેવાં વિધાનોનો ઉપયોગ કર્યો છે. અહીં "=" નિશાનીને નિરૂપક પ્રક્રિયક (Assignment Operator) કહે છે. કોઈ અચળ કિંમત કે પદાવલિના પરિણામનું ચલમાં નિરૂપણ કરવા માટે તેનો ઉપયોગ કરવામાં આવે છે.

સી ભાષામાં શોર્ટહેન્ડ (short hand) પ્રક્રિયક નામે ઓળખાતાં પ્રક્રિયકો પણ પૂરાં પાડવામાં આવ્યાં છે. ગાણિતિક પ્રક્રિયકોની પાછળ "=" નિશાની ઉમેરવાથી આ પ્રક્રિયક મેળવી શકાય છે. તેનો ઉપયોગ કરવા માટેની વાક્યરચના નીચે મુજબ છે :

variable op= constant value; અથવા *variable op= expression;*

અહીં "op" એ ગાણિતિક પ્રક્રિયક છે. જ્યારે "op="ને શોર્ટહેન્ડ પ્રક્રિયક તરીકે ઓળખવામાં આવે છે. શોર્ટહેન્ડ પ્રક્રિયકનાં કેટલાંક ઉદાહરણ નીચે આપ્યાં છે :

$first -= 1$; આ વિધાન $first = first - 1$;ને સમકક્ષ છે.

$first += 3$; આ વિધાન $first = first + 3$;ને સમકક્ષ છે.

$first *= (second + third)$; આ વિધાન $first = first * (second + third)$;ને સમકક્ષ છે.

શોર્ટહેન્ડ પ્રક્રિયક સરળ ઉપયોગિતા પૂરી પાડે છે. સરળ હોવા છતાં તેના સક્ષમ ઉપયોગ માટે થોડો મહાવરો જરૂરી છે.

સંબંધસૂચક પ્રક્રિયકો (Relational Operators)

સંબંધસૂચક પ્રક્રિયકો સમાન પ્રકારના બે ઓપરેન્ડને સરખાવવાની સુવિધા પૂરી પાડે છે અને સામાન્ય રીતે પ્રોગ્રામના અમલીકરણનો પ્રવાહ બદલવા માટે તેનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, આપણે ગ્રાહકને તેણે ખરીદેલી વસ્તુની રકમ તપાસી તે પ્રમાણે વળતર આપવા ઈચ્છીએ છીએ. જો કિંમત કોઈ નિશ્ચિત રકમથી વધુ હોય તો જ ગ્રાહકને વળતર આપવાનું છે. આ ઉદાહરણને સી ભાષામાં `if (total_purchase > 10000) discount = 500;` તરીકે રજૂ કરી શકાય.

સરખામણી કરવા માટે સી ભાષા છ સંબંધસૂચક પ્રક્રિયકો પૂરાં પાડે છે. તમામ સંબંધસૂચક પ્રક્રિયકોની યાદી અને તેના ઉપયોગો કોષ્ટક 11.5માં દર્શાવ્યા છે.

પ્રક્રિયક	ઉપયોગ
<code>==</code>	બે ઓપરેન્ડની સમાનતા (equality) ચકાસશે.
<code>!=</code>	બે ઓપરેન્ડની અસમાનતા (non equality) ચકાસશે.
<code>></code>	બે ઓપરેન્ડમાંથી મોટી (greater) કિંમત ચકાસશે.
<code><</code>	બે ઓપરેન્ડમાંથી નાની (smaller) કિંમત ચકાસશે.
<code>>=</code>	બે ઓપરેન્ડમાંથી મોટી કિંમત અથવા સમાનતા (greater or equality) ચકાસશે.
<code><=</code>	બે ઓપરેન્ડ માટે નાની કિંમત અથવા સમાનતા (smaller or equality) ચકાસશે.

કોષ્ટક 11.5 : સંબંધસૂચક પ્રક્રિયકો

અહીં એ નોંધવું જરૂરી છે કે બે ઓપરેન્ડની સમાનતા ચકાસવા માટે `"=="` ને બદલે `"= "` પ્રક્રિયકનો ઉપયોગ થાય છે. કારણ કે, સી ભાષામાં `"= "` પ્રક્રિયકનો ઉપયોગ નિરૂપક પ્રક્રિયક તરીકે કરવામાં આવે છે. સંબંધસૂચક પ્રક્રિયકોના ઉપયોગ માટે નીચેની વાક્યરચનાનો ઉપયોગ કરવામાં આવે છે :

expression-1 Rop expression-2

અહીં "Rop" એટલે સંબંધસૂચક પ્રક્રિયક (relational operator) તથા expression-1 અને expression-2 ગાણિતિક પદાવલિ, ચલ કે અચળ હોઈ શકે છે. "if" જેવા નિર્ણય માળખાં તથા "for", "while" અને "do..while" જેવા નિયંત્રણ માળખાં સાથે સંબંધસૂચક પ્રક્રિયકોનો ઉપયોગ કરવામાં આવે છે. આ માળખાઓ વિશે પ્રકરણ-13 અને 14માં વિસ્તૃત ચર્ચા કરવામાં આવી છે. સંબંધસૂચક પ્રક્રિયકનો ઉપયોગ કર્યો હોય તેવા ઘણા પ્રોગ્રામ આ પુસ્તકમાં જોઈ શકાશે.

વધારા / ઘટાડા સૂચક પ્રક્રિયકો (Increment / Decrement Operators)

પ્રોગ્રામમાં આવેલ ચલની કિંમતમાં કેટલીકવાર 1 જેટલો વધારો કે ઘટાડો કરવાની જરૂર પડતી હોય છે. સામાન્ય રીતે વિધાનોના ગણને પુનરાવર્તિત કરવાના હોય ત્યારે આ પ્રકારની સ્થિતિ ઉદ્ભવે છે. અહીં લૂપના અમલની સંખ્યાની દેખરેખ રાખે એવા ચલની જરૂર પડે છે. આ માટે શોર્ટ હેન્ડ પ્રક્રિયકો (short hand operators)નો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે આ માટે `variable += 1` અથવા `variable -= 1` વાક્યરચનાનો ઉપયોગ કરી શકાય.

સી ભાષા આ કાર્ય માટે બે વિશિષ્ટ યુનરી (unary) પ્રક્રિયકો `"++"` અને `"--"` પૂરાં પાડે છે. આ પ્રક્રિયકોને એક જ ઓપરેન્ડની જરૂર પડે છે. `"++"` ને વધારા સૂચક (Increment) પ્રક્રિયક અને `"--"` ને ઘટાડા સૂચક (Decrement) પ્રક્રિયક તરીકે ઓળખવામાં આવે છે. આ પ્રક્રિયકોના ઉપયોગ માટેની વાક્યરચના નીચે મુજબ છે :

`++ variable;` અથવા `variable ++`

`-- variable;` અથવા `variable --`

અહીં, `"++"` અને `"--"`નો ઉપયોગ ચલની આગળ કે પાછળ ઉમેરીને કરવામાં આવે છે. ચલની આગળ ઉમેરવામાં આવે ત્યારે તેને 'પ્રી-ઇન્ક્રિમેન્ટ' (pre-increment) અથવા પ્રી-ડિક્રિમેન્ટ (pre-decrement) પ્રક્રિયક તરીકે ઓળખવામાં આવે છે. અન્યથા તેને ચલની પાછળ ઉમેરવામાં આવે તો પોસ્ટ ઇન્ક્રિમેન્ટ (post increment) અથવા પોસ્ટ ડિક્રિમેન્ટ (post decrement) પ્રક્રિયક કહેવાય છે.

આ પ્રક્રિયકોના પરિણામ સ્વરૂપે ચલની કિંમતમાં 1નો વધારો કે ઘટાડો કરવામાં આવે છે. આ વધારા કે ઘટાડાની અસર પ્રોગ્રામમાં કરવામાં આવેલ પ્રક્રિયકની રીત પર અવલંબે છે. ધારો કે વિધાન નીચે મુજબ લખ્યાં છે :

```
int first = 15, second = 20, result;
```

```
result = first + second++;
```

અહીં, "result" ચલની કિંમત 35 થશે, 36 નહીં. જ્યારે બીજા વિધાનને અંતે "second" ચલની કિંમત 21 બનશે. તેની પહેલાંની કિંમતમાં 1 ઉમેરવામાં આવશે.

અહીં એ નોંધવું જરૂરી છે કે પોસ્ટ ઇન્ક્રીમેન્ટ પ્રક્રિયકનો ઉપયોગ કરવામાં આવે ત્યારે પદાવલિનું મૂલ્યાંકન કરવા માટે ચલની જૂની કિંમતનો ઉપયોગ કરવામાં આવેશે અને ત્યાર પછી ચલની કિંમતમાં 1 નો વધારો કરવામાં આવે છે.

આ જ રીતે જો નીચેનાં વિધાનો લખવામાં આવે,

```
int first = 15, second = 20, result;
```

```
result = first + ++second;
```

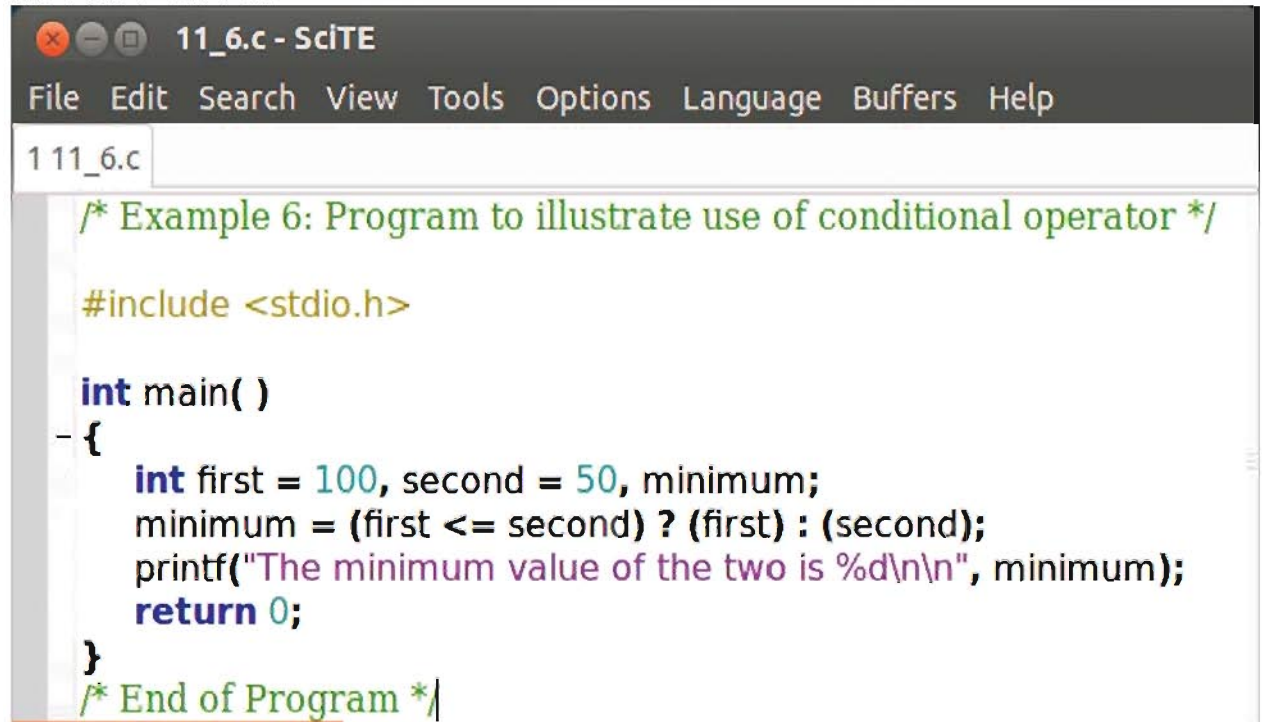
તો "result" ચલની પરિણામી કિંમત 36 હશે. જ્યારે બીજા વિધાનના અંતે "second" ચલની કિંમત 21 બનશે. અહીં એ નોંધવું જરૂરી છે કે, પ્રી-ઇન્ક્રીમેન્ટ પ્રક્રિયકનો ઉપયોગ કરવામાં આવે ત્યારે સૌ પ્રથમ ચલની કિંમતમાં 1નો વધારો કરવામાં આવે છે તથા ત્યાર પછી આ નવી કિંમતનો ઉપયોગ કરી પદાવલિનું મૂલ્યાંકન કરવામાં આવે છે.

શરતી પ્રક્રિયક (Conditional Operators)

શરતોને ચકાસવા માટે સી ભાષા ટર્નરી પ્રક્રિયક (ternary operator) અથવા શરતી પ્રક્રિયક (conditional operator) નામે ઓળખાતાં પ્રક્રિયક પૂરાં પાડે છે. આ પ્રક્રિયકને વ્યાખ્યાયિત કરવા માટે બે નિશાનીઓ- "?" : " નો ઉપયોગ કરવામાં આવે છે. શરતી પ્રક્રિયકના ઉપયોગ માટેની વાક્યરચના નીચે મુજબ છે :

(condition) ? (True statement) : (False statement);

અહીં, condition એ first < second જેવી કોઈ સંબંધસૂચક પદાવલિનો નિર્દેશ કરે છે. બે સંખ્યાઓમાંથી નાની સંખ્યા શોધવા માટેના પ્રોગ્રામની મદદથી શરતી પ્રક્રિયકનો ઉપયોગ સમજાવે. આકૃતિ 11.12 આ પ્રોગ્રામ માટેનું કોડ લિસ્ટિંગ દર્શાવે છે.



```
11_6.c - ScITE
File Edit Search View Tools Options Language Buffers Help

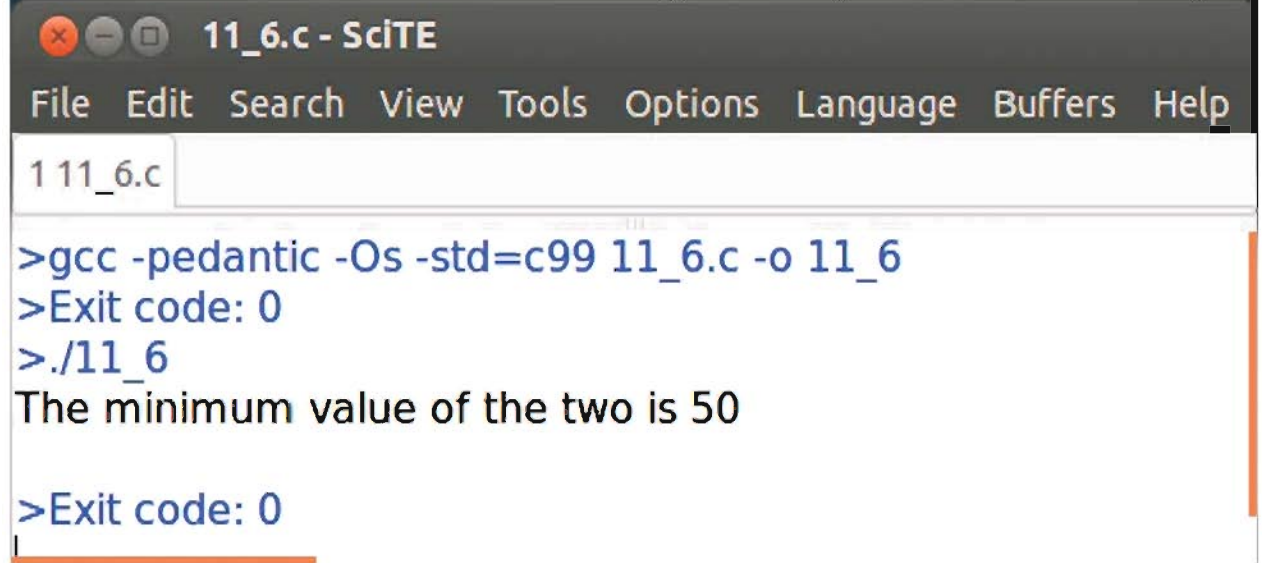
1 11_6.c
/* Example 6: Program to illustrate use of conditional operator */
#include <stdio.h>

int main( )
- {
    int first = 100, second = 50, minimum;
    minimum = (first <= second) ? (first) : (second);
    printf("The minimum value of the two is %d\n\n", minimum);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.12 : શરતી પ્રક્રિયકનો ઉપયોગ દર્શાવતો પ્રોગ્રામ

સમજૂતી (Explanation)

પ્રોગ્રામનું પ્રથમ વિધાન ત્રણ પૂર્ણાંક ચલની ઘોષણા કરી તેમાંથી બે ચલમાં કિંમત ઉમેરે છે. "first" અને "second" ચલમાં અનુક્રમે 100 અને 50 કિંમતો ઉમેરવામાં આવે છે. ત્યાર પછી શરતી પ્રક્રિયક દ્વારા ચકાસવામાં આવે છે કે "first" ચલની કિંમત "second" ચલની કિંમત કરતાં ઓછી અથવા તેના જેટલી છે કે નહીં. જો શરત સાચી (true) હોય તો "minimum" ચલને first ચલની કિંમત આપવામાં આવે છે. અન્યથા તેમાં 'second' ચલની કિંમત ઉમેરવામાં આવે છે. ત્યાર પછી યોગ્ય સંદેશ સાથે minimum ચલની કિંમત સ્ક્રીન પર દર્શાવવામાં આવે છે. અંતમાં પ્રોગ્રામમાંથી બહાર નીકળવાનો નિર્દેશ છે. ઉદાહરણ 11.6નું પરિણામ આકૃતિ 11.13માં દર્શાવવામાં આવ્યું છે.



```
11_6.c
>gcc -pedantic -Os -std=c99 11_6.c -o 11_6
>Exit code: 0
>./11_6
The minimum value of the two is 50
>Exit code: 0
```

આકૃતિ 11.13 : ઉદાહરણ 11.6નું પરિણામ

તાર્કિક પ્રક્રિયકો (Logical Operators)

કેટલીકવાર પરિણામ મેળવવા માટે એક જ સમયે એકથી વધુ શરતોની ચકાસણી કરવાની જરૂર પડે છે. જ્યારે કેટલીકવાર શરતોના ગણમાંથી કોઈ પણ એક શરત સંતોષાય તો પણ પરિણામ મેળવવું જરૂરી બને છે. આ પ્રકારના સંબંધને સામાન્ય રીતે તાર્કિક સંબંધ (logical relation) તરીકે ઓળખવામાં આવે છે. ઉદાહરણ તરીકે, બે વિષયોની પરીક્ષા આપી હોય તેવા કોઈ વિદ્યાર્થીનો વિચાર કરો. જો તે વિદ્યાર્થીએ બંને વિષયોમાં 35 કે તેથી વધુ ગુણ મેળવ્યા હશે તો જ તેને 'ઉત્તીર્ણ' જાહેર કરવામાં આવશે. અહીં આ સંબંધને 'જો (પ્રથમ વિષયના ગુણ >= 35) અને (બીજા વિષયના ગુણ >= 35) હોય તો ઉત્તીર્ણ અન્યથા અનુત્તીર્ણ' આ પ્રકારે દર્શાવી શકાય.

આ પ્રકારના સંબંધ રજૂ કરવા માટે સી ભાષા તાર્કિક પ્રક્રિયકોની સુવિધા પૂરી પાડે છે. કોષ્ટક 11.6 આ તાર્કિક પ્રક્રિયકોની યાદી અને તેના ઉપયોગો દર્શાવે છે.

પ્રક્રિયક	ઉપયોગ
&&	તાર્કિક AND
	તાર્કિક OR
!	તાર્કિક NOT

કોષ્ટક 11.6 : તાર્કિક પ્રક્રિયકો

આપેલ તમામ શરતો ફરજિયાત સંતોષવી હોય ત્યારે AND તાર્કિક પ્રક્રિયકનો ઉપયોગ કરવામાં આવે છે. તથા આપેલ તમામ શરતોમાંથી ઓછામાં ઓછી એક શરત સંતોષવી હોય ત્યારે OR તાર્કિક પ્રક્રિયકનો ઉપયોગ કરવામાં આવે છે. તાર્કિક પ્રક્રિયકોના પરિણામ સ્વરૂપે 0 અથવા 1 મળે છે. અહીં 'શૂન્ય' એ 'ખોટા' (false) તથા 1 એ 'સાચા' (true) પરિણામનો નિર્દેશ કરે છે. કોષ્ટક 11.7 જુદા જુદા તાર્કિક પ્રક્રિયકોનું પરિણામ દર્શાવે છે.

Expression 1	Expression 2	! (Expression 2)	Expression 1 && Expression 2	Expression 1 Expression 2
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

કોષ્ટક 11.7 : તાર્કિક પ્રક્રિયકોનું પરિણામ

અહીં જોઈ શકાય છે કે, જો બંને પદાવલિ '1' પરત કરે તો જ AND પ્રક્રિયક 1 પરત કરે છે. જ્યારે આપેલમાંથી કોઈ પણ એક પદાવલિ 1 પરત કરે તો OR પ્રક્રિયક 1 પરત કરે છે.

હવે, નીચે આપેલ માપદંડને આધારે વિદ્યાર્થીનો વર્ગ (Grade) શોધી આપે તે માટેનો પ્રોગ્રામ બનાવીએ :

વર્ગ (Grade)	ગુણ (Marks)
A વર્ગ	જો ટકા 70 કે તેથી વધુ હોય
B વર્ગ	જો ટકા 60 થી 69ની વચ્ચે હોય
C વર્ગ	જો ટકા 35 થી 59ની વચ્ચે હોય
F વર્ગ	જો ટકા 35 થી ઓછા હોય.

આકૃતિ 11.14માં આ પ્રોગ્રામનું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 11.15 તેનું પરિણામ દર્શાવે છે.

```

11_7.c - SciTE
File Edit Search View Tools Options Language Buffers Help
11_7.c
/* Example 7: Program to illustrate use of logical operator */

#include <stdio.h>
int main( )
{
    int percentage = 55;

    /* Verify whether percentage is between 0 and 100 using AND */
    if ( percentage >= 0 && percentage <= 100 )
    {
        /* Check whether percentage greater than 70 */
        if ( percentage >= 70 )
            printf("You have secured A grade\n\n");

        /* Check whether percentage between 60 and 69 */
        if ( percentage >= 60 && percentage < 70 )
            printf("You have secured B grade\n\n");

        /* Check whether percentage between 35 and 60 */
        if ( percentage >= 35 && percentage < 60 )
            printf("You have secured C grade\n\n");

        /* Check whether percentage less than 35 */
        if ( percentage < 35 )
            printf("You have secured F grade\n\n");
    }
    else /* Executed when percentage below 0 or above 100 */
    {
        printf("You have entered wrong value in percentage");
        printf("Please enter value between 0 and 100\n\n");
    }
    return 0;
}
/* End of Program */

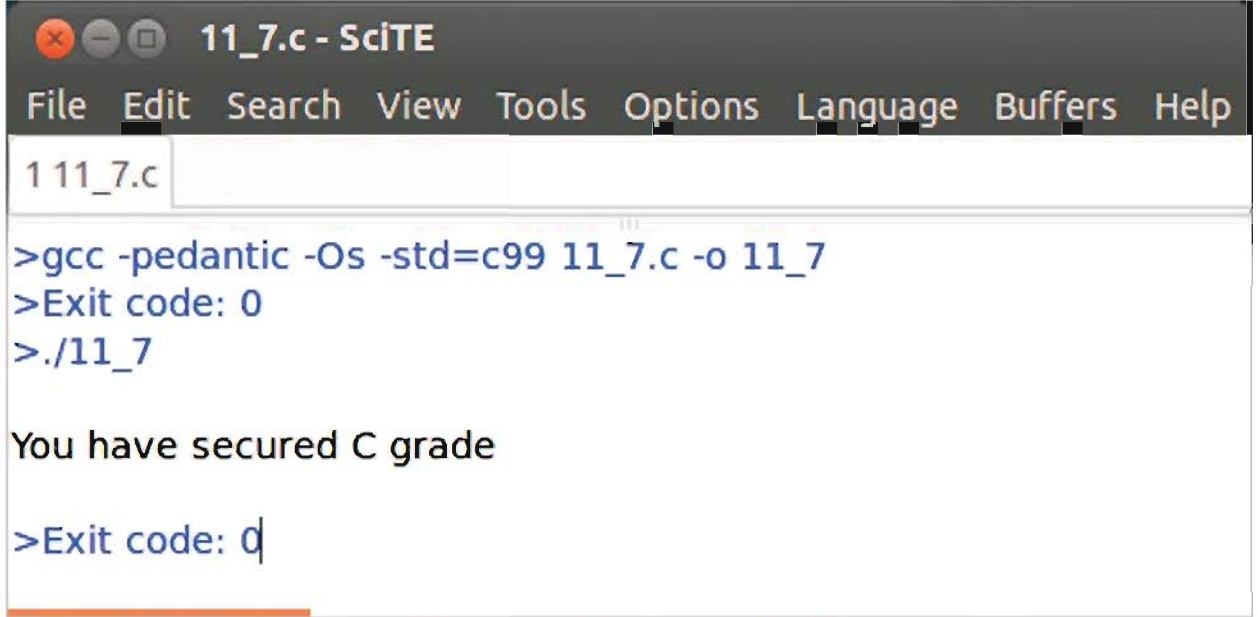
```

આકૃતિ 11.14 : તાર્કિક પ્રક્રિયકનો ઉપયોગ દર્શાવતો પ્રોગ્રામ

સમજૂતી (Explanation)

અહીં પ્રથમ વિધાન દ્વારા "percentage" નામના પૂર્ણાંક ચલને ઘોષિત કરી તેમાં કિંમત આપવામાં આવે છે. શરૂઆતમાં "percentage" ચલની કિંમત 0 થી 100 વચ્ચે આપેલ છે કે નહીં તે ચકાસવામાં આવે છે. જો જવાબ 'ના' મળે તો 'else' વિભાગમાં આવેલ સંદેશ "You have entered wrong value in percentage, Please enter value between 0 and 100" દર્શાવી પ્રોગ્રામમાંથી બહાર નીકળી શકાય છે.

જો સરખામણીનું પરિણામ 'હા' (true) મળશે તો ટકાની કિંમતને if વિધાન અને તાર્કિક પ્રક્રિયકો દ્વારા ચકાસવામાં આવશે. જો if વિધાનમાં આપેલ માપદંડનું પરિણામ હકારાત્મક મળશે તો તેનો સંદેશ સ્ક્રીન પર દર્શાવવામાં આવશે. આકૃતિ 11.15 જુઓ.



```
11_7.c
>gcc -pedantic -Os -std=c99 11_7.c -o 11_7
>Exit code: 0
>./11_7

You have secured C grade

>Exit code: 0
```

આકૃતિ 11.15 : ઉદાહરણ 11.7નું પરિણામ

બિટવાઈઝ પ્રક્રિયકો (Bitwise Operators)

આપણે આ પ્રકરણમાં જોયું કે ડેટાને મેમરીસ્થાન પર બિટ (bit) સ્વરૂપે સંગ્રહવામાં આવે છે. સીધા જ બિટ સ્તરે કાર્ય કરવા માટે સી ભાષામાં બિટવાઈઝ પ્રક્રિયકોની સુવિધા આપવામાં આવી છે. સી ભાષામાં ઉપલબ્ધ બિટવાઈઝ પ્રક્રિયકોની યાદી અને તેના ઉપયોગ કોષ્ટક 11.8માં આપવામાં આવ્યા છે.

પ્રક્રિયક	ઉપયોગ
&	Bitwise AND
	Bitwise OR
~	Bitwise NOT
^	Bitwise Exclusive OR
<<	આપેલ બિટ જેટલા અંકોને ડાબી બાજુ ખસેડવા
>>	આપેલ બિટ જેટલા અંકોને જમણી બાજુ ખસેડવા

કોષ્ટક 11.8 : બિટવાઈઝ પ્રક્રિયકો (Bitwise Operators)

બિટવાઈઝ AND, OR, Exclusive OR (XOR), Left Shift અને Right Shift પ્રક્રિયકોનો ઉપયોગ કરવા માટે બે ઓપરેન્ડ જરૂરી છે. બીજી તરફ NOT પ્રક્રિયક માત્ર એક ઓપરેન્ડ સાથે કાર્ય કરે છે. અહીં ઓપરેન્ડ 0 અને 1 કિંમતો સ્વરૂપે હોય છે. આ પ્રક્રિયકોની વિસ્તૃત ચર્ચા પુસ્તકની મર્યાદા બહાર છે.

વિશિષ્ટ પ્રક્રિયકો (Special Operators)

સી ભાષા `sizeof()`, `","`, `","`, `","`, `","`, `","` અને `"&"` અને `"*"` જેવાં વિશિષ્ટ પ્રક્રિયકો પૂરાં પાડે છે. આ વિભાગમાં આપણે માત્ર `","` અને `sizeof()` પ્રક્રિયકોની ચર્ચા કરીશું. `","` (અલ્પવિરામ) પ્રક્રિયકનો ઉપયોગ ઘણી જગ્યાએ કરવામાં આવે છે. અલ્પવિરામ (comma) પ્રક્રિયકનો ઉપયોગ નિર્ણય માળખાં અને નિયંત્રણ માળખાંમાં પણ કરવામાં આવે છે જેની ચર્ચા હવે પછીના પ્રકરણોમાં કરવામાં આવી છે.

ઘટકનો સંગ્રહ કરવા માટે જરૂરી બાઈટની સંખ્યા પરત કરવા માટે `sizeof()` નામના એક વિશિષ્ટ પ્રક્રિયકનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, `size = sizeof(int)` વિધાન `size` ચલમાં `4` કિમતનો સંગ્રહ કરશે. કારણ કે `int` ડેટા પ્રકાર મેમરીની 4 બાઈટ જેટલી જગ્યાનો ઉપયોગ કરે છે.

અલ્પવિરામ (comma) પ્રક્રિયક અને `sizeof()` પ્રક્રિયકનો ઉપયોગ દર્શાવતો પ્રોગ્રામ આકૃતિ 11.16માં આપેલ છે તથા તેનું પરિણામ 11.17માં દર્શાવવામાં આવ્યું છે.

```
11_8.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_8.c
/* Example 8: Program to illustrate use of comma and size of operator */
#include <stdio.h>
int main( )
- {
    float first, second, result;
    int size;
    /* Use of comma operator */
    result = (first = 125, second = 15, first / second);
    printf("\nFirst = %f, Second = %f, Result = %f", first, second, result);

    /* Use of sizeof operator */
    size = sizeof(result);
    printf("\nSize allocated by compiler to variable result is %d bytes\n\n", size);
}
/* End of Program */
```

આકૃતિ 11.16 : વિશિષ્ટ પ્રક્રિયકોનો ઉપયોગ દર્શાવતો પ્રોગ્રામ

```
11_8.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_8.c
>gcc -pedantic -Os -std=c99 11_8.c -o 11_8
>Exit code: 0
>./11_8

First = 125.000000, Second = 15.000000, Result = 8.333333
Size allocated by compiler to variable result is 4 bytes

>Exit code: 0
```

આકૃતિ 11.17 : ઉદાહરણ 11.8નું પરિણામ

સમજૂતી (Explanation)

અહીં, પ્રથમ વિધાન ત્રણ અપૂર્ણ (float) ચલની ઘોષણા કરે છે તથા બીજું વિધાન એક પૂર્ણાંક ચલને ઘોષિત કરે છે. અમલ કરી શકાય તેવા ત્રીજા વિધાનમાં અલ્પવિરામ (comma) પ્રક્રિયકનો ઉપયોગ કરી પદાવલિ લખવામાં આવી છે. આપેલ વિધાન આ મુજબ કાર્ય કરશે : પ્રથમ જમણી બાજુની પદાવલિનું મૂલ્યાંકન કરવામાં આવશે. પ્રથમ પ્રક્રિયામાં "first" ચલને 125 કિંમત આપવામાં આવશે, બીજી પ્રક્રિયામાં "second" ચલને 15 કિંમત આપવામાં આવશે. અને ત્રીજી પ્રક્રિયામાં "first" નો "second" વડે ભાગાકાર કરવામાં આવશે. અંતમાં, ભાગાકારનું પરિણામ "result" ચલમાં સંગ્રહવામાં આવશે. ત્યાર પછીનું વિધાન "first", "second" અને "result" ચલની કિંમત દર્શાવે છે. ત્યાર પછીના વિધાનમાં "result" ચલમાં આવેલ કિંમતને મેમરીમાં આપવામાં આવેલી જગ્યાનો "size" નામના ચલમાં સંગ્રહ કરવામાં આવશે.

પદાવલિનું મૂલ્યાંકન (Evaluation of Expression)

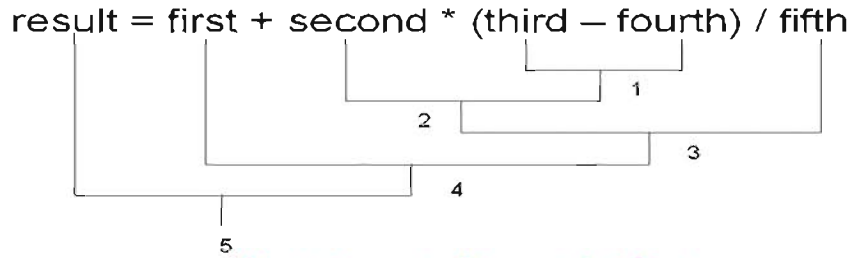
અત્યાર સુધીમાં આપણે સી ભાષામાં ઉપલબ્ધ પ્રક્રિયકોનો અભ્યાસ કરી તેનો પદાવલિમાં કેવી રીતે ઉપયોગ કરવો તે શીખ્યા. હવે, પદાવલિનું મૂલ્યાંકન કેવી રીતે કરવામાં આવે છે તે જોઈએ. નીચેના વિધાનમાં રહેલ પદાવલિ જુઓ :

$result = first + second * third - fourth;$

અહીં, પ્રથમ 'second' અને 'third' ચલની કિંમતોનો ગુણાકાર કરવામાં આવશે. આ ગુણાકારની કિંમતને ત્યાર પછી first ચલની કિંમતમાં ઉમેરવામાં આવશે અને આ મધ્યવર્તી કિંમતમાંથી fourthની કિંમત બાદ કરવામાં આવશે. જો પદાવલિમાં એક જ અગ્રતા ધરાવતા બે પ્રક્રિયકોનો ઉપયોગ કરવામાં આવ્યો હશે તો તેમનું મૂલ્યાંકન ડાબીથી જમણી તરફ કરવામાં આવશે. કૌંસનો ઉપયોગ કરી આ ક્રમને બદલી શકાય છે.

$result = first + second * (third - fourth) / fifth;$

હવે અહીં third - fourth નું મૂલ્યાંકન સૌપ્રથમ કરવામાં આવશે, મધ્યવર્તી કિંમતનો second ચલની કિંમત સાથે ગુણાકાર કરવામાં આવશે, પછી fifth ચલની કિંમત વડે ભાગાકાર કરવામાં આવશે અને અંતમાં first ચલની કિંમતમાં તેને ઉમેરવામાં આવશે. આકૃતિ 11.18 જુઓ.



આકૃતિ 11.18 : પદાવલિના મૂલ્યાંકનનો ક્રમ

પ્રક્રિયકોની અગ્રતા (Priority of Operators)

અત્યાર સુધી જોયેલા પ્રક્રિયકોને તેમની નિશ્ચિત અગ્રતા આપવામાં આવી છે. ઉદાહરણ $result = first + second * (third - fourth) / fifth;$ વિધાનમાં જોઈ શકાય છે કે * ને +, - કે / કરતાં વધુ અગ્રતા આપવામાં આવી હતી. સી ભાષાના સર્જકોએ આ અગ્રતા પૂર્વનિશ્ચિત કરેલી છે. આમાંના કેટલાંક પ્રક્રિયકોની અગ્રતા કોષ્ટક 11.9માં દર્શાવી છે. પરિશિષ્ટ III માં આ માટેની વિસ્તૃત સમજ આપવામાં આવી છે.

Operator	Operation Used for	Associativity	Precedence
()	Function call	Left to Right	First
[]	Array expression		
++	Increment	Right to Left	Second
--	Decrement		
sizeof()	Size of operand		
*	Multiplication	Left to Right	Third
/	Division		
%	Modulo division		
+	Binary addition	Left to Right	Fourth
-	Binary subtraction		

કોષ્ટક 11.9 : પ્રક્રિયકોની અગ્રતા

પ્રકાર બદલી (Type Conversion)

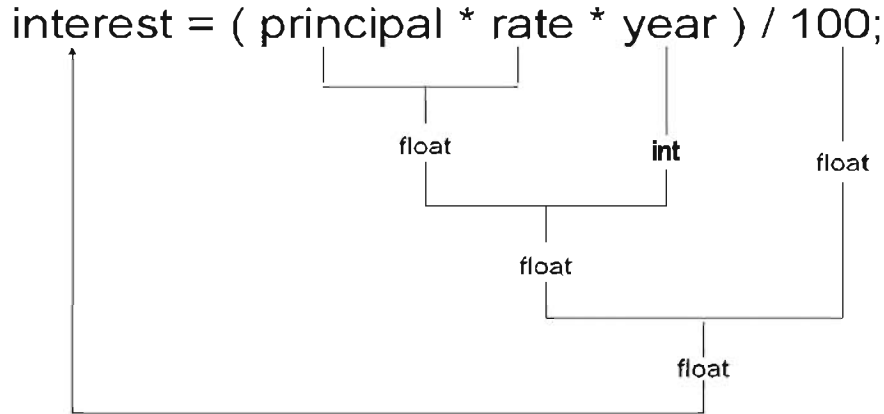
પદાવલિમાં આવેલ તમામ ઓપરેન્ડ એક જ પ્રકારના હોય તો જ તેનું મૂલ્યાંકન કરવામાં આવે છે. આ લાક્ષણિકતાને કારણે પદાવલિના મૂલ્યાંકન માટે તેમાં આવેલા ડેટા પ્રકારની આંતરિક બદલી જરૂરી બને છે. ઉદાહરણ તરીકે નીચેનો કોડ જુઓ :

```
int year=2;
```

```
float principal, rate, interest;
```

```
interest = (principal * rate * year) / 100;
```

આ પદાવલિનું મૂલ્યાંકન આકૃતિ 11.19માં દર્શાવ્યા મુજબ કરવામાં આવશે :



આકૃતિ 11.19 : પ્રકાર બદલી

આંતરિક ફેરબદલીને ઓવરવ્રાઇડ (override) થવા માટે ટાઇપ કાસ્ટિંગ (type casting) નામની પ્રક્રિયાનો ઉપયોગ કરી શકાય છે. કોઈ પણ કિંમતનો પ્રકાર બદલવા (type cast) માટે નીચે આપેલ વાક્યરચનાનો ઉપયોગ કરવામાં આવે છે.

(data type) variable અથવા (datatype) constant

ટાઇપ કાસ્ટિંગનો ઉપયોગ કરી બનાવવામાં આવેલા પ્રોગ્રામનું કોડ લિસ્ટિંગ આકૃતિ 11.20 માં આપેલ છે. આકૃતિ 11.21 પ્રોગ્રામનું પરિણામ દર્શાવે છે.

```
11_9.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 11_9.c
/* Example 9: Program to illustrate use of type cast */

#include <stdio.h>
int main( )
- {
    int total_cost = 575 , quantity = 24;
    float cost_item;

    /* C expression for calculating cost per item without type casting */
    cost_item = total_cost / quantity;

    printf("\nCost of one item without type casting is %f", cost_item);

    /* C expression for calculating cost per item without type casting */
    cost_item = total_cost / (float) quantity;

    printf("\nCost of one item after type casting is %f\n\n", cost_item);
    return 0;
}
/* End of Program */
```

આકૃતિ 11.20 : ટાઇપ કાસ્ટિંગ દર્શાવતો પ્રોગ્રામ

```
11_9.c - SciTE
File Edit Search View Tools Options Language Buffers Help
1 11_9.c
>gcc -pedantic -Os -std=c99 11_9.c -o 11_9
>Exit code: 0
>./11_9

Cost of one item without type casting is 23.000000
Cost of one item after type casting is 23.958334

>Exit code: 0
```

આકૃતિ 11.21 : ઉદાહરણ 11.9નું પરિણામ

સમજૂતી (Explanation)

આ પ્રોગ્રામમાં 24 વસ્તુઓની કિંમત આપેલી છે અને વસ્તુની એકમ કિંમત શોધવાની છે. "total_cost" અને "quantity" ચલને પૂર્ણાંક તથા "cost_item" ચલને અપૂર્ણાંક તરીકે વ્યાખ્યાયિત કરવામાં આવ્યા છે. એ જોઈ શકાય છે કે બંને ઓપરેન્ડ પૂર્ણાંક હોવાને કારણે પ્રથમ સી પદાવલિ વસ્તુની એકમ કિંમત તરીકે 23.000000 દર્શાવે છે. આ જ પદાવલિનો જ્યારે ટાઈપ કાસ્ટિંગ સાથે ઉપયોગ કરવામાં આવે છે ત્યારે તે ઈચ્છિત પરિણામ 23.958334 દર્શાવે છે.

સંગ્રાહક વર્ગો (Storage Classes)

સી પ્રોગ્રામમાં ઉપયોગમાં લેવામાં આવતા ચલનો સંગ્રહ સામાન્ય રીતે કમ્પ્યુટરની પ્રાઈમરી મેમરીમાં કરવામાં આવે છે. કેટલાક ચલનો રજિસ્ટરમાં સંગ્રહ કરવો પણ શક્ય છે. સી ભાષા automatic, external, register, અને static નામના ચાર સંગ્રાહક વર્ગો (storage classes) પૂરાં પાડે છે. આ સંગ્રાહક વર્ગો દ્વારા ચલનો સંગ્રહ કરવા માટેનું સ્થાન નિશ્ચિત કરવામાં આવે છે. તમામ સંગ્રાહક વર્ગોની એક પછી એક ચર્ચા કરીએ.

ઓટોમેટિક ચલ (Automatic Variables)

તમામ ચલનો પૂર્વ નિર્ધારિત સંગ્રાહક વર્ગ 'ઓટોમેટિક' હોય છે. ચલને 'ઓટોમેટિક' સ્વરૂપમાં નિશ્ચિત રૂપે (explicitly) વ્યાખ્યાયિત કરવા માટે auto કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. ચલ માટે ઓટોમેટિક સંગ્રાહક વર્ગ વ્યાખ્યાયિત કરવા માટેની વાક્યરચના નીચે મુજબ છે :

auto data type identifier;

ઉદાહરણ તરીકે, auto int number; વ્યાખ્યા દર્શાવે છે કે ચલ number એ પૂર્ણ પ્રકારનો છે અને તેનો સંગ્રાહક વર્ગ ઓટોમેટિક છે. જ્યારે કોઈ વિધેયનો અમલ કરવામાં આવે છે (called) ત્યારે આ પ્રકારના ચલની રચના કરવામાં આવે છે તથા જ્યારે આ વિધેય બોલાવનાર (caller) વિધેયને નિયંત્રણ પરત મોકલે છે ત્યારે આ પ્રકારના ચલનો નાશ કરવામાં આવે છે. આ પ્રકારના ચલને શરૂઆતમાં પૂર્વનિર્ધારિત રીતે અનિશ્ચિત કિંમત (garbage value - એવી કિંમત કે ઉપયોગકર્તા માટે જેનું કોઈ મહત્વ નથી) આપવામાં આવે છે અને તેનો પ્રાઈમરી મેમરીમાં સંગ્રહ કરવામાં આવે છે.

એક્સ્ટર્નલ ચલ (External Variable)

કેટલીકવાર પ્રોગ્રામરને કોઈ ચલ બે વિધેય કે પ્રોગ્રામ વચ્ચે વહેંચવાની જરૂર પડે છે. બે જુદા જુદા વિધેય કે પ્રોગ્રામ વચ્ચે ચલની વહેંચણી માટે 'એક્સ્ટર્નલ' સંગ્રાહક વર્ગનો ઉપયોગ કરી શકાય છે. આ માટે એ જરૂરી છે કે ઓછામાં ઓછા એક પ્રોગ્રામમાં તે ચલ વૈશ્વિક (global) ચલ તરીકે ઘોષિત થયેલો હોવો જોઈએ. નીચેની વાક્યરચના દ્વારા ચલ માટે એક્સ્ટર્નલ સંગ્રાહક વર્ગ વ્યાખ્યાયિત કરી શકાય.

extern datatype identifier;

ઉદાહરણરૂપે, extern char choice; વ્યાખ્યા દર્શાવે છે કે choice ચલ અક્ષર પ્રકારનો છે અને તેનો સંગ્રાહક વર્ગ એક્સ્ટર્નલ છે. એક્સ્ટર્નલ ચલની પૂર્વ નિર્ધારિત કિંમત શૂન્ય છે અને તેનો પ્રાઈમરી મેમરીમાં સંગ્રહ કરવામાં આવે છે.

રજિસ્ટર ચલ (Register Variable)

ઝડપી ઉપયોગ કરવા માટે ચલને CPU રજિસ્ટરમાં સંગ્રહી શકાય છે. સી ભાષા આ માટે 'register' નામનો સંગ્રાહક વર્ગ પૂરો પાડે છે. નીચેની વાક્યરચના દ્વારા ચલ માટે રજિસ્ટર સંગ્રાહક વર્ગ વ્યાખ્યાયિત કરી શકાય છે :

register datatype identifier;

ઉદાહરણ રૂપે, `register int counter;` વ્યાખ્યા દર્શાવે છે કે `counter` ચલ પૂર્ણાંક પ્રકારનો છે અને તેનો સંગ્રાહક વર્ગ રજિસ્ટર છે. પૂર્વ નિર્ધારિત રીતે આ પ્રકારના ચલમાં અનિશ્ચિત કિંમત (garbage value) નો ઉપયોગ કરવામાં આવે છે.

સ્ટેટિક ચલ (Static Variable)

જ્યારે ચલનો 'સ્ટેટિક' સંગ્રાહક વર્ગ સાથે ઉપયોગ કરવામાં આવે છે ત્યારે નિશ્ચિત વિસ્તાર માટે તે ચલની કિંમત કાયમી બનાવી શકાય છે. સ્ટેટિક સંગ્રાહક વર્ગ ધરાવતા ચલની કિંમતને પ્રાથમિક મેમરીમાં કાયમી ધોરણે સાચવવામાં આવે છે તથા પૂર્વનિર્ધારિત રીતે તેમાં શૂન્ય કિંમત મૂકવામાં આવે છે. ચલનો સ્ટેટિક સંગ્રાહક વર્ગ વ્યાખ્યાયિત કરવા માટે નીચે આપેલ વાક્યરચનાનો ઉપયોગ કરવામાં આવે છે :

static datatype identifier;

ઉદાહરણ રૂપે, `static int max;` વ્યાખ્યા દર્શાવે છે કે `max` ચલ પૂર્ણાંક પ્રકારનો છે અને તેનો સંગ્રાહક વર્ગ સ્ટેટિક છે.

સારાંશ

આ પ્રકરણમાં આપણે મૂળભૂત ડેટા પ્રકારો `int`, `float`, `char` અને `void` વિશે અભ્યાસ કર્યો. `typedef`, `enum` અને `array` જેવા ઉપયોગકર્તા દ્વારા નિર્મિત અને તારવેલા ડેટા પ્રકારો વિશે પણ માહિતી મેળવી. નિરૂપક પ્રક્રિયક દ્વારા ચલમાં કેવી રીતે કિંમત ઉમેરવી તે પણ આપણે જોયું. ત્યાર પછી આપણે ગાણિતિક, સંબંધસૂચક, વધારા-ઘટાડા સૂચક, શરતી, તાર્કિક, બિટવાઈઝ અને વિશિષ્ટ પ્રક્રિયકો વિશે અભ્યાસ કર્યો. જુદા જુદા પ્રક્રિયકોનો ઉપયોગ કરી પદાવલિની રચના કરતાં તથા તેનું મૂલ્યાંકન કરતાં પણ આપણે શીખ્યા. અંતમાં સી ભાષા દ્વારા પૂરા પાડવામાં આવતા ચાર સંગ્રાહક વર્ગો વિશે માહિતી મેળવી.

શિક્ષકોને સૂચના (Instruction for Teachers)

અહીં ચર્ચવામાં આવેલા તમામ ડેટા પ્રકારો ANSI C99 ધારાધોરણ આધારિત છે. અહીં આપવામાં આવેલાં ઉદાહરણ પ્રક્રિયકોને સમજવા માટે છે. શિક્ષક વધુ સ્વાધ્યાય બનાવી વિદ્યાર્થીઓને આપી શકે છે.

સ્વાધ્યાય

1. ડેટા પ્રકાર એટલે શું? તેનું મહત્ત્વ જણાવો.
2. `void` ડેટા પ્રકારનું મહત્ત્વ જણાવો.
3. ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત ડેટા પ્રકારનું મહત્ત્વ જણાવો.
4. તારવેલા ડેટા પ્રકારનું મહત્ત્વ જણાવો.
5. નીચેના વિધાનો ખરાં છે કે ખોટાં તે જણાવો :
 - (a) `char` ડેટા પ્રકાર માટે બે બાઈટ જગ્યા જરૂરી છે.
 - (b) ચિહ્નરહિત (unsigned) પૂર્ણાંક ચલ શૂન્યથી ઓછી સંખ્યાનો સંગ્રહ કરી શકે છે.
 - (c) `double` ડેટા પ્રકારનો ઉપયોગ પૂર્ણાંક સંખ્યાઓનો સંગ્રહ કરવા માટે છે.
 - (d) `int` ડેટા પ્રકારનો વિસ્તાર વધારવા માટે તેની આગળ `long` કી-વર્ડ ઉમેરવામાં આવે છે.
 - (e) અપૂર્ણાંક સંખ્યાઓનું વૈજ્ઞાનિક સ્વરૂપ `exponent * 10 mantissa` તરીકે રજૂ કરવામાં આવે છે.

6. આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

- (1) પૂર્ણાંક ડેટા પ્રકાર માટે નીચેનામાંથી કયા કી-વર્ડનો ઉપયોગ કરવામાં આવે છે ?
(a) integer (b) Integer (c) INTEGER (d) int
- (2) નીચેનામાંથી કયો વિકલ્પ ખાલી કિંમતનો નિર્દેશ કરે છે ?
(a) void (b) Void (c) char (d) float
- (3) gcc કંપાઈલર દ્વારા float પ્રકારને કયા કદની મેમરી આપવામાં આવે છે ?
(a) 1 (b) 2 (c) 4 (d) 6
- (4) સી ભાષામાં એક અક્ષર વ્યાખ્યાયિત કરવા માટે નીચેનામાંથી કયો કી-વર્ડ યોગ્ય છે ?
(a) char (b) character (c) CHAR (d) CHARACTER
- (5) સી ભાષામાં નીચેનામાંથી કયો ડેટા પ્રકાર ઉપયોગકર્તા દ્વારા નિર્મિત છે ?
(a) int (b) enum (c) char (d) float
- (6) સી ભાષામાં આવેલા ડેટા પ્રકારને ઉપનામ આપવા માટે નીચેનામાંથી કયા કી-વર્ડનો ઉપયોગ કરવામાં આવે છે ?
(a) enum (b) def (c) pointer (d) typedef
- (7) સી ભાષામાં "&" નિશાની કયા પ્રકારનો પ્રક્રિયક છે ?
(a) સંબંધસૂચક (b) ગણિતિક (c) તાર્કિક (d) બિટવાઈઝ
- (8) સી ભાષામાં "!" નિશાની કયા પ્રકારનો પ્રક્રિયક છે ?
(a) સંબંધસૂચક (b) ગણિતિક (c) તાર્કિક (d) બિટવાઈઝ
- (9) સી ભાષામાં "=" નિશાનીનો ઉપયોગ કયા પ્રક્રિયક તરીકે કરવામાં આવે છે ?
(a) સમાનતા (b) નિરૂપણ (c) નોંધ (d) સમીકરણ
- (10) સી ભાષામાં value++ નો અર્થ શું કરી શકાય ?
(a) પોસ્ટ ઇન્ક્રિમેન્ટ (b) પોસ્ટ ડિક્રિમેન્ટ (c) પ્રી-ઇન્ક્રિમેન્ટ (d) પ્રી ડિક્રિમેન્ટ

પ્રાયોગિક સ્વાધ્યાય

1. આપેલ મીટરને મિલીમીટરમાં ફેરવવા માટેનો સી પ્રોગ્રામ લખો.
2. $c = (f-32) * 5/9$ સમીકરણનો ઉપયોગ કરી આપેલ ફેરનહીટને સેલ્સિયસમાં ફેરવવા માટેનો સી પ્રોગ્રામ લખો.
3. ત્રણ સંખ્યાઓની સરેરાશ શોધવા માટેનો સી પ્રોગ્રામ લખો.
4. 2000 ચોમી ના વિસ્તારમાં 50 ચો સેમીની એક એવી કુલ કેટલી લાદી જડી શકાય તે શોધવા માટેનો પ્રોગ્રામ લખો.
5. લંબચોરસનું ક્ષેત્રફળ શોધવા માટેનો સી પ્રોગ્રામ લખો.
6. આપેલ સંખ્યા અન્ય આપેલ સંખ્યા સાથે નિઃશેષ ભાજ્ય છે કે નહીં તે શોધવા માટેનો સી પ્રોગ્રામ લખો.
7. ₹ P ની લોન R% વ્યાજના દર સાથે N વર્ષો માટે લેવામાં આવી હોય તો તે પરથી ચક્રવૃદ્ધિ વ્યાજ શોધવા માટેનો સી પ્રોગ્રામ લખો. P, R અને Nની કિંમતો વિદ્યાર્થી દ્વારા આપવામાં આવશે.
8. ધનનું ધનફળ શોધવા માટેનો સી પ્રોગ્રામ લખો.
9. વર્તુળનું ક્ષેત્રફળ શોધવા માટેનો સી પ્રોગ્રામ લખો.
10. ₹ P ની લોન R% વ્યાજના દર સાથે N વર્ષો માટે લેવામાં આવી હોય તો તે પરથી સાદું વ્યાજ શોધવા માટેનો સી પ્રોગ્રામ લખો. P, R અને Nની કિંમતો વિદ્યાર્થી દ્વારા આપવામાં આવશે.



નિવેશ (Input) / નિર્ગમ (Output) પ્રક્રિયાઓનો ઉપયોગ

તમામ પ્રોગ્રામિંગ ભાષાઓ નિવેશ કરવામાં આવેલી વિગતો વાંચવાની સુવિધા આપે છે તેને ઇનપુટ ઓપરેશન (input operation) કહે છે તથા પરિણામ દર્શાવવાની સુવિધા આપે છે જેને આઉટપુટ ઓપરેશન (output operation) તરીકે ઓળખવામાં આવે છે. અહીં 'નિવેશ' (input) એટલે કી-બોર્ડ જેવા કોઈ ઇનપુટ સાધન તરફથી મળેલ ડેટા વાંચવો તથા આ 'નિર્ગમ' (output) એટલે મોનિટર, પ્રિન્ટર જેવા આઉટપુટ સાધન પર ડેટાને દર્શાવવો. આ ઇનપુટ અને આઉટપુટ I/O પ્રક્રિયાઓ (I/O operations) નામથી વધુ પ્રચલિત છે. ઇનપુટ અને આઉટપુટ ક્રિયાઓના અમલ માટે સી ભાષા પાસે કોઈ આંતરસ્થાપિત વિધાન નથી. ડેટાના ઇનપુટ અને આઉટપુટ માટેની પ્રક્રિયાઓ પ્રમાણભૂત ઇનપુટ/આઉટપુટ આંતરપ્રસ્થાપિત લાઈબ્રેરી પ્રોગ્રામ દ્વારા પાર પાડવામાં આવે છે. આ પ્રકરણમાં આપણે `getchar()`, `getch()`, `gets()`, `scanf()`, `printf()`, `putchar()`, `putc()`, `puts()` વગેરે વિધેય દ્વારા I/O પ્રક્રિયાઓ કેવી રીતે પાર પાડી શકાય તે વિશે ચર્ચા કરીશું. આપણે સુગ્રંથિત (formatted) ઇનપુટ અને આઉટપુટ ક્રિયાઓ વિશે પણ અભ્યાસ કરીશું.

પ્રોગ્રામમાં ડેટા પર પ્રક્રિયા કરવા માટે ચલનો ઉપયોગ કરવામાં આવે છે. ચલને ઇનપુટ આપવા માટેની મુખ્ય બે રીત છે. એક રીતમાં એસાઈનમેન્ટ ઓપરેટર (assignment operator) નો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, `number=5`; વિધાન `number` ચલમાં 5 સંખ્યાનો સંગ્રહ કરશે. બીજી રીત મુજબ પ્રોગ્રામનો અમલ કરતી વખતે ઉપયોગકર્તાએ આપેલ ડેટા વાંચવામાં આવશે. આ માટે ઇનપુટ પ્રક્રિયા માટેના કોઈ આંતરસ્થાપિત વિધેયનો ઉપયોગ કરી શકાય છે. હવે આપણે સામાન્ય રીતે ઉપયોગમાં લેવામાં આવતા આંતરસ્થાપિત ઇનપુટ વિધેય વિશે ચર્ચા કરીએ.

આંતરસ્થાપિત ઇનપુટ વિધેય (Inbuilt input function)

કી-બોર્ડ, માઉસ વગેરે જેવાં ઇનપુટ સાધનો દ્વારા પ્રોગ્રામને ડેટા ઇનપુટ કરવો શક્ય છે. સી ભાષા ઇનપુટ સંબંધિત ઘણાં વિધેય પૂરાં પાડે છે, જેનો સી લાઈબ્રેરી દ્વારા સંગ્રહ થયેલો હોય છે. સી લાઈબ્રેરીમાં આવેલા કોઈ પણ આંતરસ્થાપિત વિધેયનો ઉપયોગ કરવા માટે, `#include` વિધાનનો ઉપયોગ કરી પ્રોગ્રામની શરૂઆતમાં જે-તે લાઈબ્રેરી ફાઈલ ઉમેરવી જરૂરી છે. આપણે ઘણા પ્રોગ્રામમાં પ્રોગ્રામની શરૂઆતમાં નીચેના વિધાનોનો ઉપયોગ કર્યો હતો તે તમે નોંધ્યું હશે :

`#include<stdio.h>`

`stdio.h`નો અર્થ છે standard input-output header ફાઈલ. ઇનપુટ અને આઉટપુટ પ્રક્રિયાઓને સંબંધિત ઘણા વિધેયનો સમાવેશ આ હેડર ફાઈલમાં કરવામાં આવ્યો છે. `#include` વિધાન `stdio.h` ફાઈલને શોધી તેની વિગતોને પ્રોગ્રામની શરૂઆતમાં ઉમેરવાની કમ્પાઈલરને સૂચના આપે છે. ત્યારપછી આ હેડર ફાઈલની વિગતો પ્રોગ્રામનો એક ભાગ બની જાય છે. સી લાઈબ્રેરી વિધેય સંબંધિત વધુ ચર્ચા સી વિધેયના પ્રકરણમાં કરવામાં આવી છે. અહીં એ નોંધ જરૂરી છે કે, જ્યારે `scanf()` વિધેયનો ઉપયોગ કરવામાં આવે ત્યારે કેટલાક કમ્પ્યૂટરોમાં આ હેડર ફાઈલ ઉમેરવી જરૂરી હોતી નથી. હવે નીચેના વિભાગમાં `getchar()`, `getch()`, `getc()` અને `gets()` વિધેય વિશે ચર્ચા કરીએ.

getchar()

સી ભાષામાં પ્રોગ્રામનો અમલ કરતી વખતે એક અક્ષર વાંચવા માટેનો સૌથી સરળ માર્ગ `getchar()` વિધેયનો ઉપયોગ છે. `getchar()` વિધેયનું પ્રમાણભૂત સ્વરૂપ નીચે દર્શાવ્યું છે :

`variable_name = getchar();`

`variable_name` એ `char` ડેટાટાઈપ ધરાવતો સી ભાષામાં માન્ય કોઈ પણ ચલ છે. `getchar()` વિધેયને કોઈ પ્રાચલ (parameter) આપવામાં આવતા નથી. જ્યારે તેનો અમલ કરવામાં આવે ત્યારે તે એક અક્ષર વાંચે છે. આ વિધેય `int` પરત કરે છે, જે આપેલ અક્ષરનો આસ્કી કોડ હોય છે, પરંતુ આપણે ઉપયોગકર્તાએ આપેલ ઇનપુટનો `char` ચલમાં સંગ્રહ કરી શકીએ છીએ.

હવે આપણે ઉદાહરણ 12.1નો ઉપયોગ કરી `getchar()` વિધેયના ઉપયોગને સમજવાનો પ્રયત્ન કરીએ. અહીં આપણે સંદેશ દર્શાવવા માટે ઉપયોગકર્તા દ્વારા લખવામાં આવેલ ઉત્તરનો ઉપયોગ કરવા ઇચ્છીએ છીએ. ઉદાહરણ 12.1નું કોડ લિસ્ટિંગ આકૃતિ 12.1માં આપવામાં આવ્યું છે તથા આકૃતિ 12.2 તેનું પરિણામ દર્શાવે છે.

```
9-1.c - SciTE
File Edit Search View Tools Options Language Buffers Help
9-1.c
/* Example 1: Program to demonstrate use of getchar() function. */

#include<stdio.h>
int main()
{
    char answer;

    printf("Would you like to study? \n");
    printf("Enter 'Y' for yes and 'N' for no: ");

    answer = getchar();

    if(answer == 'Y' || answer == 'y')
        printf("Go to study room...\n");
    else
        printf("Go to play ground...\n");

    return 0;
}
// End of Program
```

આકૃતિ 12.1 : ઉદાહરણ 12.1નું કોડ લિસ્ટિંગ

```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 9-1.c
kpp@ubuntu:~$ ./a.out
Would you like to study?
Enter 'Y' for yes and 'N' for no: Y
Go to study room...
kpp@ubuntu:~$
```

આકૃતિ 12.2 : ઉદાહરણ 12.1નું પરિણામ

સમજૂતી (Explanation)

ઉદાહરણ 12.1નું પ્રથમ વિધાન એક અક્ષરનો સંગ્રહ કરવા માટે `answer` નામના ચલને ઘોષિત કરે છે. `getchar()` વિધેયનો અમલ કરવામાં આવે ત્યારે પ્રોગ્રામ ઉપયોગકર્તા દ્વારા કોઈ કી દબાવવાની પ્રતીક્ષા કરે છે. ઉપયોગકર્તાએ ઉમેરેલો અક્ષર `answer` ચલમાં સંગ્રહવામાં આવે છે અને તે અક્ષર સ્ક્રીન ઉપર પણ દર્શાવવામાં આવે છે. ઉદાહરણ તરીકે, જો ઉપયોગકર્તા 'Y' કે 'y' ઉમેરે તો "Go to study room..." સંદેશ દર્શાવવામાં આવશે. જો ઉપયોગકર્તા અન્ય કોઈપણ અક્ષર ઉમેરે તો "Go to playground..." સંદેશ દર્શાવવામાં આવશે.

getch()

getch() વિધેયનો ઉપયોગ પણ એક અક્ષર મેળવવા માટે કરી શકાય છે. getch() અને getchar() વચ્ચેનો તફાવત એ છે કે, અહીં ઉમેરવામાં આવેલો અક્ષર સ્ક્રીન પર દર્શાવવામાં આવતો નથી. ઉપયોગકર્તા દ્વારા ટાઇપ કરવામાં આવેલ અક્ષર સ્ક્રીન પર દર્શાવવો ન હોય ત્યારે આ વિધેયનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ 12.1માં ઉમેરેલ getch() વિધેયને getch() સાથે બદલી બંને વિધેય વચ્ચેના તફાવતનું અવલોકન કરો.

getc()

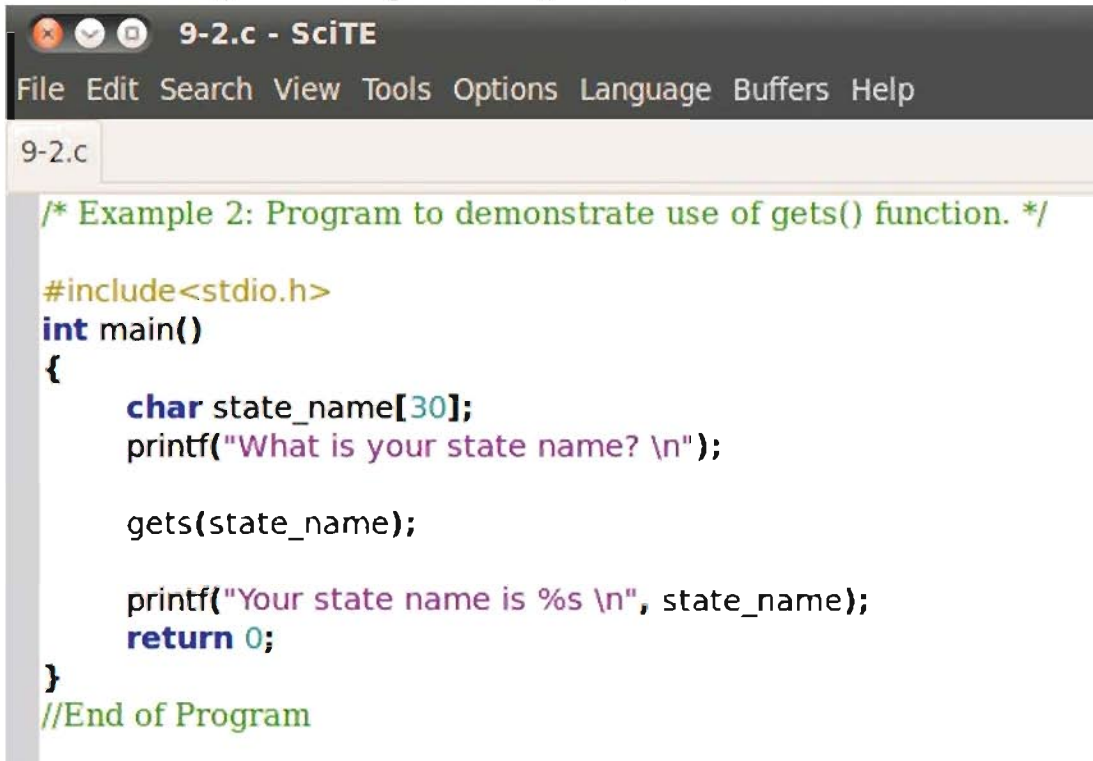
getchar() અને getc(), વિધેયની જેમ getc() વિધેયનો ઉપયોગ પણ એક અક્ષર મેળવવા માટે કરવામાં આવે છે. પરંતુ અહીં તફાવત એ છે કે getc() પ્રમાણભૂત ઇનપુટ સાધનને બદલે ફાઈલમાં રહેલ અક્ષર વાંચે છે. getc() સંબંધિત વધુ ચર્ચા આ પાઠ્યક્રમની મર્યાદા બહાર છે.

gets()

એક સમયે એક જ અક્ષર મેળવી શકાય તે માટેના getchar(), getch() અને getc() વિધેયનો આપણે અભ્યાસ કર્યો. બે અક્ષરો મેળવવા હોય તો getch() વિધેયનો બે વખત અમલ કરી શકાય. પરંતુ અક્ષરોનું જૂથ મેળવવાની જરૂર હોય ત્યારે getch() વિધેયનો અનેકવાર અમલ કરવો એ યોગ્ય તર્ક નથી. અક્ષરોની હારમાળા (string) મેળવવા માટે gets() વિધેયનો ઉપયોગ વધુ સારો વિકલ્પ છે. gets() વિધેયની સામાન્ય વાક્યરચના નીચે મુજબ છે :

gets(variable_name);

gets() વિધેય ક્રિમત તરીકે એક સ્ટ્રિંગ સ્વીકારે છે. અહીં variable_name એ અક્ષરોનો એરે (character array) છે. અક્ષરોના એરે વિશેની વધુ ચર્ચા ‘એરે’ પ્રકરણમાં કરવામાં આવી છે. પ્રોગ્રામના અમલીકરણ દરમિયાન gets() વિધેયનો અમલ કરવામાં આવે ત્યારે તે ઇનપુટ સાધનનો ઉપયોગ કરી ઉપયોગકર્તાએ ઉમેરેલા અક્ષરોની પ્રતીક્ષા કરે છે. ઉપયોગકર્તા દ્વારા એન્ટર (new line character) દબાવવામાં ન આવે ત્યાં સુધી અક્ષરો મેળવવામાં આવે છે અને ત્યાર પછી તેના અંત ભાગમાં ‘નલ’ અક્ષર ('\0') ઉમેરી સ્ટ્રિંગ પૂર્ણ કરવામાં આવે છે. આ વિધેયના અમલ બાદ તમામ અક્ષરો અને અંતમાં રહેલ નલ ક્રિમતનો variable_name ચલમાં સંગ્રહ કરવામાં આવે છે. ઉદાહરણ 12.2નો ઉપયોગ કરી gets() વિધેયની કાર્યપદ્ધતિ સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 12.3માં ઉદાહરણ 12.2નું કોડ-લિસ્ટિંગ આપવામાં આવેલું છે તથા આકૃતિ 12.4 તેનું પરિણામ દર્શાવે છે.



```
9-2.c - SciTE
File Edit Search View Tools Options Language Buffers Help

9-2.c

/* Example 2: Program to demonstrate use of gets() function. */

#include<stdio.h>
int main()
{
    char state_name[30];
    printf("What is your state name? \n");

    gets(state_name);

    printf("Your state name is %s \n", state_name);
    return 0;
}
//End of Program
```

આકૃતિ 12.3 : ઉદાહરણ 12.2નું કોડ-લિસ્ટિંગ


```
kpp@ubuntu: ~  
File Edit View Terminal Help  
kpp@ubuntu:~$ ./a.out  
What is your state name?  
Gujarat  
Your state name is Gujarat  
kpp@ubuntu:~$
```

આકૃતિ 12.4 : ઉદાહરણ 12.2નું પરિણામ

ઉદાહરણ 12.2માં ઉપયોગકર્તાએ ઉમેરેલ ઇનપુટને state_name ચલમાં સંગ્રહવામાં આવશે. આ જ સ્ટ્રિંગને printf() વિધેય દ્વારા સ્ક્રીન પર દર્શાવવામાં આવશે. printf() વિધાન વિશે વિસ્તૃત ચર્ચા આ પ્રકરણમાં આગળ ઉપર કરવામાં આવી છે.

અહીં એ નોંધ લેવી જરૂરી છે કે, gets() વિધેય ધરાવતા કોઈ પણ પ્રોગ્રામના કમ્પાઈલેશન વખતે "the gets function is dangerous and should not be used" એ પ્રકારનો ચેતવણી-સંદેશ દર્શાવવામાં આવે છે. પરંતુ આ માત્ર ચેતવણી-સંદેશ હોવાથી પ્રોગ્રામનું અમલીકરણ ચાલુ રાખી શકાય છે.

સુગ્રંથિત નિવેશ (Formatted input)

કેટલીકવાર પ્રોગ્રામમાં એવી સ્થિતિ ઉદ્ભવે છે જ્યારે અત્યાર સુધીમાં ચર્ચેલા ઇનપુટ વિધેય વિશેષ ઉપયોગી બનતાં નથી. getch(), getch() અને getc() વિધેયનો ઉપયોગ એક અક્ષર મેળવવા માટે કરવામાં આવે છે તથા ઉપયોગકર્તા એન્ટર કી ન દબાવે ત્યાં સુધી અનેક અક્ષરો મેળવવા માટે gets() વિધેયનો ઉપયોગ કરી શકાય છે. પરંતુ ઉપયોગકર્તા જો '33 Mudra Ahmedabad' આ પ્રકારનો ડેટા ઉમેરવા ઇચ્છતો હોય તો ?

અહીં ડેટાના પ્રથમ વિભાગમાં વિદ્યાર્થીનો અનુક્રમ તથા બીજા અને ત્રીજા વિભાગમાં અનુક્રમે વિદ્યાર્થીનું નામ અને શહેરનું નામ દર્શાવ્યું છે. અહીં એ જોઈ શકાય છે કે અનુક્રમ એ પૂર્ણાંક પ્રકારનો ડેટા છે તથા બાકીના બે ડેટામાં અક્ષરોનો સમાવેશ કરવામાં આવ્યો છે.

આ પ્રકારનો ડેટા મેળવવા માટે સી ભાષા સુગ્રંથિત નિવેશ (formatted input) વિધેય નામની સુવિધા પૂરી પાડે છે. એક ફોર્મેટેડ ઇનપુટ વિધેય scanf()નો ઉપયોગ આપણે ઘણા પ્રોગ્રામોમાં કર્યો છે. scanf() વિધેય દ્વારા ફોર્મેટેડ ઇનપુટ માટે ઉપલબ્ધ વિવિધ વિકલ્પોનો અભ્યાસ કરીએ.

scanf()

scanf()નો અર્થ છે, scan formatted. આ વિધેય int, char, float વગેરે જેવા નિશ્ચિત સ્વરૂપમાં રહેલા ડેટા ઉમેરવાની સુવિધા આપે છે. scanf() વિધેય માટેની સામાન્ય વાક્યરચના નીચે આપેલી છે :

scanf("control string", &variable1, &variable2, &variableN);

જે ડેટા સ્વરૂપે ચલની કિંમતોનો સંગ્રહ કરવાનો હોય તેને કન્ટ્રોલ સ્ટ્રિંગ દ્વારા સ્પષ્ટ કરવામાં આવે છે. variable1, variable2, ... variableN એ ચલનાં નામ છે. ચલના નામની આગળ &(ampersand) નિશાની ઉમેરવામાં આવે છે તથા દરેક નામ અલ્પવિરામ (,) થી છૂટાં પાડવામાં આવે છે. સી ભાષામાં & નિશાનીને address of પ્રક્રિયક તરીકે ઓળખવામાં આવે છે. ઉપયોગકર્તા દ્વારા ઉમેરવામાં આવેલા ઇનપુટનો સંગ્રહ કરવા માટેનું સ્થાન તેના દ્વારા સ્પષ્ટ કરવામાં આવે છે. આ આકૃતિ 12.5 દ્વારા સમજી શકાશે.

Variable Name → rollno

Value →

33

Memory Address → 2345

આકૃતિ 12.5 : મેમરીની સંરચના (Memory Layout)

આકૃતિ 12.5માં દર્શાવ્યા મુજબ rollno ચલનું નામ છે તથા તેની કિંમત 33 છે. ચલની કિંમતનો વાસ્તવિક સંગ્રહ 2345 મેમરીસ્થાન પર થયો છે. (આ સંખ્યા માત્ર ઉદાહરણ છે; તે કમ્પ્યુટરનું કોઈ પણ મેમરીસ્થાન હોઈ શકે છે.) સી ભાષામાં દરેક ચલને તેનાં નામ અને મેમરીસ્થાન દ્વારા વ્યાખ્યાયિત કરવામાં આવે છે. જ્યારે પ્રોગ્રામમાં ચલના નામનો ઉલ્લેખ કરવામાં આવે ત્યારે સી કમ્પાઈલર પ્રક્રિયા માટે ચલના મેમરીસ્થાનનો ઉપયોગ કરે છે. address of પ્રક્રિયકની વધુ ચર્ચા આ પુસ્તકની મર્યાદા બહાર છે. સી પોઈન્ટરનો અભ્યાસ કરતી વખતે તમે આ અભિગમ શીખશો.

કન્ટ્રોલ સ્ટ્રિંગને ‘ફોર્મેટ સ્ટ્રિંગ’ (format string) તરીકે પણ ઓળખવામાં આવે છે. ઉપયોગકર્તાએ આપેલ ડેટા ઈનપુટનો અનુવાદ કરવા માટે તે કમ્પાઈલરને મદદરૂપ બને છે. ઈનપુટ દરમિયાન ચલની કિંમતોનો ક્રમ સ્પષ્ટ કરવાનું કાર્ય પણ કન્ટ્રોલ સ્ટ્રિંગનું છે. દરેક ફોર્મેટમાં ડેટાટાઈપને સ્પષ્ટ કરતા અક્ષરની આગળ ‘%’ નિશાની ઉમેરવામાં આવે છે. એક ઉદાહરણની મદદથી કન્ટ્રોલ સ્ટ્રિંગના ઉપયોગની સમજ મેળવીએ.

પૂર્ણાંક સંખ્યા વાંચવી (Reading Integers)

ઉપયોગકર્તા પાસેથી ત્રણ પૂર્ણાંક સંખ્યાઓ મેળવીને તેનો marks1, marks2 અને marks3 નામના ચલમાં સંગ્રહ કરવા માટે નીચે આપેલ પ્રોગ્રામખંડનો ઉપયોગ કરી શકાય.

```
int marks1, marks2, marks3;
```

```
scanf("%d %d %d", &marks1, &marks2, &marks3);
```

ઉપરના વિધાનના અમલ દરમિયાન જો ઉપયોગકર્તા 70 80 90 ઉમેરશે તો marks1 ચલમાં 70, marks2 ચલમાં 80 અને marks3 ચલમાં 90 સંખ્યાનો સંગ્રહ કરવામાં આવશે. અહીં %dની સાથે ફિલ્ડનું કદ (field width) પણ નીચે જણાવ્યા મુજબ સ્પષ્ટ કરી શકાય છે :

```
scanf("%2d %4d", &marks1, &marks2);
```

અગાઉનાં scanf() વિધાનના અમલ દરમિયાન ઉપયોગકર્તા જો 70 1234 સંખ્યાઓ ઉમેરે તો marks1માં 70 અને marks2માં 1234 સંખ્યાઓનો સંગ્રહ કરવામાં આવશે.

પરંતુ ઉપરના scanf() વિધાન માટે ઉપયોગકર્તા જો 1234 70 સંખ્યા ઉમેરશે તો marks1 ચલમાં 12 સંખ્યાનો સંગ્રહ કરવામાં આવશે. (કારણ કે, ફિલ્ડનું કદ %2d છે) તથા marks2 ચલમાં 34 સંખ્યાનો સંગ્રહ કરવામાં આવશે (જે 1234 સંખ્યાનો નહીં વંચાયેલ ભાગ છે). આ scanf() વિધાન માટે 70 કિંમત નિરર્થક બની રહેશે.

અપૂર્ણાંક સંખ્યા વાંચવી (Reading Real Numbers)

વિદ્યાર્થીએ મેળવેલ ટકા (percentage) જેવી અપૂર્ણાંક કિંમત પ્રોગ્રામમાં મેળવવા માટે નીચે આપેલ પ્રોગ્રામ-ખંડનો ઉપયોગ કરી શકાય :

```
float per1, per2;
```

```
scanf("%f %f", &per1, &per2);
```

અહીં per1 અને per2 નામના બે અપૂર્ણ ચલ વ્યાખ્યાયિત કર્યા છે. scanf() વિધાન આ ચલમાં અપૂર્ણ સંખ્યાઓનો સંગ્રહ કરે છે. આ કોડના અમલ દરમિયાન જો ઉપયોગકર્તા 85.25 90.65 સંખ્યાઓ ઉમેરે તો 85.25 સંખ્યાનો per1 ચલમાં અને 90.65 સંખ્યાનો per2 ચલમાં સંગ્રહ કરવામાં આવશે. scanf() વિધેય અપૂર્ણ સંખ્યાઓ વાંચવા માટે “%f”નો ઉપયોગ કરે છે. જો ઉપયોગકર્તા પાસેથી double ડેટાટાઈપ ધરાવતી સંખ્યા મેળવવાની હોય તો “%lf”ના સ્થાને “%Lf”નો ઉપયોગ કરવો જોઈએ.

અક્ષર અને શબ્દનું વાચન (Reading character and word)

getchar() અને gets() જેવા આંતરસ્થાપિત વિધેયોના ઉપયોગથી એક અક્ષર અને અક્ષરોનો સમૂહ (સ્ટ્રિંગ) મેળવી શકાય છે તે આપણે જોઈએ. અક્ષર અને શબ્દ વાંચવાનું આ જ કાર્ય scanf() વિધાન દ્વારા પણ અનુક્રમે %c અને %s સ્પેસિફાયરની મદદથી કરી શકાય છે. નીચેનો પ્રોગ્રામ-ખંડ જુઓ :

```
char c1, c2;
```

```
char city[20];
```

```
scanf("%c %c %s", &c1, &c2, city);
```

આપેલ કોડના અમલ દરમિયાન ઉપયોગકર્તા જો **A B Gandhinagar** જેવો ડેટા ઉમેરે તો, c1 ચલમાં 'A', c2 ચલમાં 'B' અને city નામના અક્ષરના ઓરેમાં 'Gandhinagar' ક્રિમતનો સંગ્રહ કરવામાં આવશે. અક્ષરોની ઓરે સંબંધિત વધુ ચર્ચા આ પુસ્તકના પ્રકરણ 15માં કરવામાં આવી છે.

અહીં એ નોંધ લેવી જરૂરી છે કે scanf() વિધાન દ્વારા સ્ટ્રિંગ મેળવતી વખતે ચલનાં નામ સાથે & (ampersand) નિશાની જરૂરી નથી. એ પણ યાદ રાખવું જરૂરી છે કે ઇનપુટમાં ખાલી જગ્યા આવે ત્યારે %s સ્પેસિફાયર ઇનપુટને અટકાવે છે. નીચેનું ઉદાહરણ જુઓ:

```
char city[20];
```

```
scanf("%s", city);
```

આ કોડના અમલ દરમિયાન ઉપયોગકર્તા જો **New Delhi** ઉમેરશે તો city નામના ચલમાં માત્ર "New" શબ્દનો સંગ્રહ કરવામાં આવશે. %s સ્પેસિફાયર New શબ્દ પછી આપેલી ખાલી જગ્યાને કારણે ઇનપુટને અટકાવશે. અને "Delhi" શબ્દને scanf() દ્વારા અવગણવામાં આવશે.

%[characters] અને %[^ characters]ની મદદથી મર્યાદિત ઇનપુટ

સી ભાષામાં આવેલ scanf() વિધેય %[characters]ના ઉપયોગ દ્વારા એક અદ્ભુત સુવિધા પૂરી પાડે છે, જેના ઉપયોગથી ઇનપુટ સ્ટ્રિંગમાં માત્ર સ્વીકાર્ય હોય તેવા જ અક્ષરો દર્શાવી શકાય છે. %[characters]માં આવેલ ન હોય તેવો કોઈ પણ અક્ષર ઉમેરવામાં આવે તો આવો અક્ષર પ્રથમ વખત ઉમેરવાની ઘટના સ્ટ્રિંગને અટકાવે છે. ઉદાહરણ 12.3નો ઉપયોગ કરી આ અભિગમ સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 12.6માં ઉદાહરણ 12.3નું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 12.7 તેનું પરિણામ દર્શાવે છે.

```

9-3.c - SciTE
File Edit Search View Tools Options Language Buffers Help

9-3.c

/* Example 3 Program to demonstrate use of %[characters] specification. */

#include<stdio.h>
int main()
{
    char student_name[30];

    printf("Enter your name containing only alphabets and space: \n");
    scanf("%[a-z A-Z]", student_name);
    printf("Your name stored in variable is %s \n", student_name);
    return 0;
}
//End of Program

```

આકૃતિ 12.6 : ઉદાહરણ 12.3નું કોડ-લિસ્ટિંગ

```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 9-3.c
kpp@ubuntu:~$ ./a.out
Enter your name containing only alphabets and space:
Ragi Bharat Patel
Your name stored in variable is Ragi Bharat Patel
kpp@ubuntu:~$ ./a.out
Enter your name containing only alphabets and space:
Ragi B. Patel
Your name stored in variable is Ragi B
kpp@ubuntu:~$ 

```

આકૃતિ 12.7 : ઉદાહરણ 12.3નું પરિણામ

ઉદાહરણ 12.3ના `scanf()` વિધાનમાં આપેલ `%[a-zA-Z]` દર્શાવે છે કે માત્ર `a` થી `z`, ખાલી જગ્યા અને `A` થી `Z` અક્ષરોનો જ સ્વીકાર કરી `student_name` નામના ચલમાં તેને સંગૃહીત કરવામાં આવશે. માટે, પ્રથમ અમલ દરમિયાન વિદ્યાર્થીના નામનો ચલમાં સંગ્રહ કરવામાં આવ્યો છે. પરંતુ બીજા અમલ દરમિયાન ઉપયોગકર્તા પૂર્ણવિરામ (.) ઉમેરે છે, જે યાદીમાં આવેલ નથી. તેથી ચલમાં માત્ર "Raghi B"નો જ સંગ્રહ કરવામાં આવે છે.

જો ઉપયોગકર્તા યાદીમાં આપેલ અક્ષરો પૈકી કોઈ પણ અક્ષર ઉમેરશે તો `scanf()`માં `%[^characters]` સ્વરૂપનો ઉપયોગ ઇનપુટ પ્રક્રિયાને તત્કાલ અટકાવી દેશે. ઉદાહરણ 12.4નો ઉપયોગ કરી આ અભિગમ સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 12.8માં ઉદાહરણ 12.4નું કોડ-લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 12.8 તેનું પરિણામ દર્શાવે છે.

```

9-4.c - SciTE
File Edit Search View Tools Options Language Buffers Help

9-4.c
/* Example 4: Program to demonstrate use of %[ ^characters] specification. */

#include<stdio.h>
int main()
{
    char user_data[100];

    printf("Enter your details:\n");
    printf("Enter * to exit:\n");
    scanf(" %[ ^*] ", user_data);
    printf("Your details stored in variable is \n%s \n", user_data);
    return 0;
}
//End of Program

```

આકૃતિ 12.8 : ઉદાહરણ 12.4નું કોડ-લિસ્ટિંગ

```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 9-4.c
kpp@ubuntu:~$ ./a.out
Enter your details:
Enter * to exit:
My name is Purva.
My mother's name is Krina*
Your details stored in variable is
My name is Purva.
My mother's name is Krina
kpp@ubuntu:~$ 

```

આકૃતિ 12.9 : ઉદાહરણ 12.4નું પરિણામ

ઉદાહરણ 12.4ના અમલ દરમિયાન કોઈ પણ અક્ષર (નવી લીટી માટેના 'એન્ટર' સહિત) ઉમેરી શકાય છે. જ્યારે * ઉમેરવામાં આવે ત્યારે ઇનપુટ સ્ટ્રિંગને સ્વીકારવાની ક્રિયા અટકાવવામાં આવે છે. અહીં એ નોંધ લેવી જરૂરી છે કે `user_data` ચલની લંબાઈ 100 ઘોષિત કરી હોવાથી આપણા ઉદાહરણમાં માત્ર 100 અક્ષરો ઉમેરી શકાય.

મિશ્ર ડેટાનું વાચન (Reading mixed data)

એક જ `scanf()` વિધાનની મદદથી જુદા જુદા પ્રકારનો ડેટા (જેમ કે પૂર્ણાંક, અપૂર્ણાંક, અક્ષર, સ્ટ્રિંગ) ઉમેરવો શક્ય છે. આ પ્રકારના કિસ્સામાં ખાતરી કરી લેવી જોઈએ કે `scanf()`ના કન્ટ્રોલ સ્પેસિફિકેશન સાથે ઉમેરવામાં આવેલ ડેટાનો ક્રમ અને ડેટાપ્રકાર અચૂકપણે સમાન હોય. આ અભિગમ સમજવા માટે નીચે આપેલ પ્રોગ્રામ-ખંડ જુઓ :

```
int roll_no;
char grade, name[30];
float percen;
scanf("%d %f %s %c", &roll_no, &percen, name, &grade);
```

આ કોડના અમલ દરમિયાન જો **11 90.55 Vani A** વિગતો ઉમેરવામાં આવે તો `roll_no` ચલમાં 11, `percen` ચલમાં 90.55, `name` ચલમાં 'Vani' અને `grade` ચલમાં 'A' કિંમતોનો સંગ્રહ કરવામાં આવશે. પરંતુ આ જ પ્રોગ્રામ કોડ માટે જો ઉપયોગકર્તા **11 Vani A 90.55** વિગતો ઉમેરે તો કમ્પ્યુટર ભૂલનો સંદેશ દર્શાવશે કારણ કે કમ્પાઈલરને જ્યાં અપૂર્ણાંક સંખ્યાની અપેક્ષા હતી ત્યાં ઉપયોગકર્તાએ અક્ષર ઉમેર્યો છે. આ કિસ્સામાં `scanf()` વિધાન પ્રથમ કિંમત મેળવ્યા બાદ વાંચવાની પ્રક્રિયા અટકાવશે.

જુદા જુદા પ્રકારના ડેટા મેળવવા માટે `scanf()`ની કન્ટ્રોલ સ્ટ્રિંગમાં ઉપયોગમાં લઈ શકાય તેવા અક્ષરોની યાદી કોષ્ટક 12.1માં આપવામાં આવી છે.

ડેટાપ્રકાર	સંલગ્ન અક્ષર
દશાંકી પૂર્ણાંક વાંચવા માટે	%d
અક્ષર વાંચવા માટે	%c
અપૂર્ણાંક વાંચવા માટે	%f અથવા %e અથવા %g
સ્ટ્રિંગ વાંચવા માટે	%s
ચિહ્નરહિત (અનસાઈન્ડ) પૂર્ણાંક વાંચવા માટે	%u
ટૂંકા (શોર્ટ) પૂર્ણાંક વાંચવા માટે	%h
લાંબા (લોન્ગ) પૂર્ણાંક વાંચવા માટે	%ld
ડબલ સંખ્યા વાંચવા માટે	%lf
લોન્ગ ડબલ સંખ્યા વાંચવા માટે	%L

કોષ્ટક 12.1 : `scanf()`ની કન્ટ્રોલ સ્ટ્રિંગમાં ઉપયોગમાં લેવામાં આવતા અક્ષરો

આંતરપ્રસ્થાપિત આઉટપુટ વિધેય (Inbuilt output function)

કમ્પ્યુટર સિસ્ટમનાં ત્રણ મૂળભૂત કાર્યો છે : નિવેશ (input), પ્રક્રિયા (process) અને નિર્ગમ (output). ઉપયોગકર્તા પાસેથી ડેટા મેળવવા માટે ઉપલબ્ધ આંતરપ્રસ્થાપિત ઈનપુટ વિધેય વિશે આપણે જોઈએ. હવે, પ્રક્રિયામાંથી પસાર થયેલો ડેટા દર્શાવવા માટેના આંતરપ્રસ્થાપિત આઉટપુટ વિધેય વિશે અભ્યાસ કરીએ. પરિણામને મોનિટર, પ્રિન્ટર અને ફાઈલ જેવાં આઉટપુટ સાધનો પર મોકલવામાં આવે છે. સી ભાષામાં પ્રમાણભૂત આઉટપુટ સાધન મોનિટર પર પરિણામ કેવી રીતે દર્શાવી શકાય તેની ચર્ચા કરીશું. પરંતુ ફાઈલ અને પ્રિન્ટર પર પરિણામ મોકલવા અંગેની ચર્ચા આપણે નહીં કરીએ કારણ કે તે આપણા પાઠ્યક્રમમાં સમાવિષ્ટ નથી.

ઈનપુટની જેમ જ, સી લાઈબ્રેરી વિધેય દ્વારા આઉટપુટ સાધન પર પરિણામ મોકલવું શક્ય છે. `<stdio.h>` હેડર ફાઈલમાં આંતરપ્રસ્થાપિત આઉટપુટ વિધેય ઉપલબ્ધ છે. હવે `putchar()`, `puts()` અને `printf()` જેવા આઉટપુટ સંબંધિત આંતરપ્રસ્થાપિત વિધેય વિશે ચર્ચા કરીએ.

putchar()

પ્રમાણભૂત આઉટપુટ સાધન પર એક અક્ષર દર્શાવવા માટે putchar() વિધેયનો ઉપયોગ કરી શકાય છે. putchar()ની સામાન્ય વાક્યરચના નીચે મુજબ છે:

```
putchar(character);
```

અહીં character એ char પ્રકારનો ચલ અથવા તો સી ભાષામાં યોગ્ય એવો કોઈ પણ અક્ષર હોઈ શકે. જ્યારે આ વિધેયનો અમલ કરવામાં આવે છે ત્યારે મોનિટર પર અક્ષર દર્શાવવામાં આવશે. ઉદાહરણ 12.5નો ઉપયોગ કરી આ અભિગમ સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 12.10માં ઉદાહરણ 12.5નું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે જ્યારે આકૃતિ 12.11 તેનું પરિણામ દર્શાવે છે.

```
9-5.c - SciTE
File Edit Search View Tools Options Language Buffers Help
9-5.c
/* Example 5: Program to demonstrate the use of putchar() function. */
#include<stdio.h>
int main()
{
    char grade;

    printf("Enter your grade:\n");
    grade=getchar();    //reads a single character from user
    printf("Your grade is ");
    putchar(grade);    //displays a single character to screen
    printf(". \n");
    return 0;
}
//End of Program
```

આકૃતિ 12.10 : ઉદાહરણ 12.5નું કોડ લિસ્ટિંગ

```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 9-5.c
kpp@ubuntu:~$ ./a.out
Enter your grade:
A
Your grade is A.
kpp@ubuntu:~$
```

આકૃતિ 12.11 : ઉદાહરણ 12.5નું પરિણામ

આ પ્રોગ્રામ getchar() વિધેયનો ઉપયોગ કરી એક અક્ષર મેળવશે અને putchar() વિધેયનો ઉપયોગ કરી સ્ક્રીન પર એક અક્ષર દર્શાવશે.

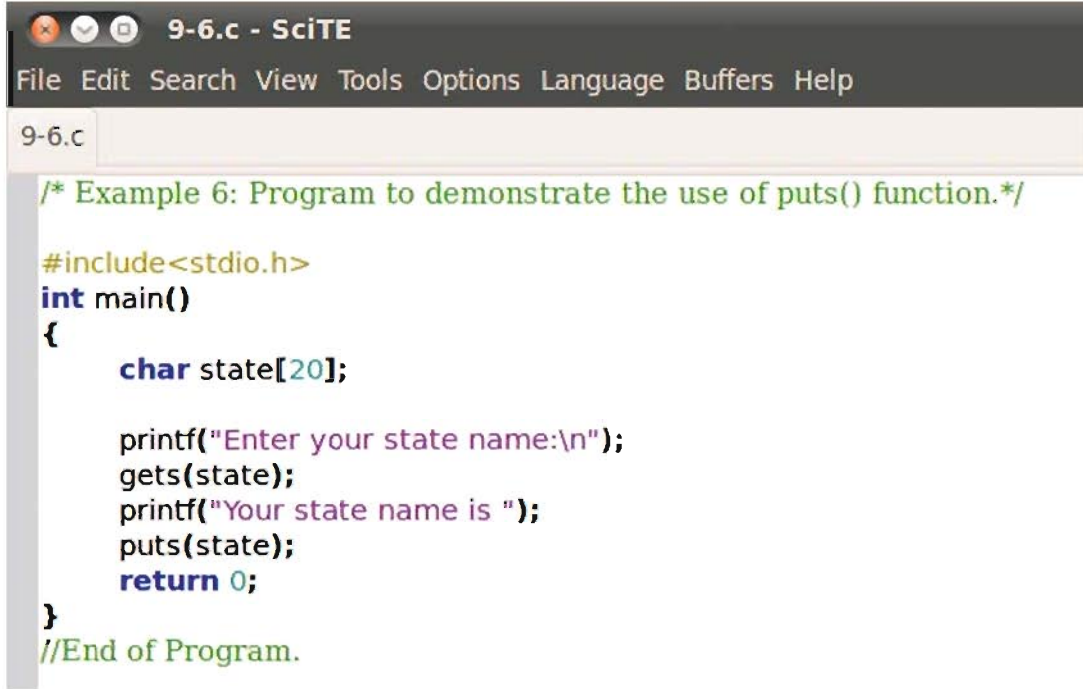
putchar() વિધેયની એક મર્યાદા એ છે કે, તે એક સમયે એક જ અક્ષર દર્શાવી શકે છે. એકથી વધુ અક્ષરો દર્શાવવા માટે સી ભાષાના લૂપ અભિગમનો ઉપયોગ કરવો પડે છે. એકથી વધુ અક્ષરો દર્શાવવા માટેનો સરળ ઉકેલ puts() વિધેયનો ઉપયોગ કરવાનો છે.

puts()

આઉટપુટ સાધન પર એકથી વધુ અક્ષરો (સ્ટ્રિંગ અથવા અક્ષરોનો એરે) દર્શાવવા માટે puts() વિધેયનો ઉપયોગ કરી શકાય છે. puts()ની સામાન્ય વાક્યરચના નીચે મુજબ છે:

puts(variable_name);

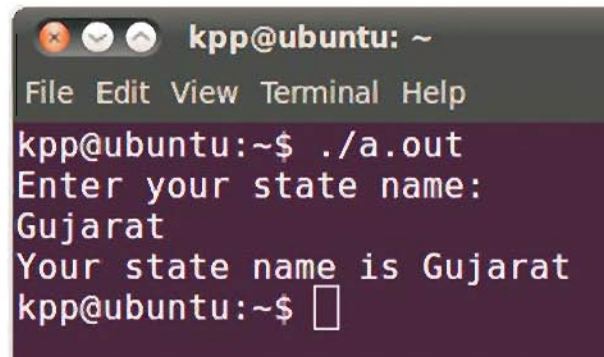
અહીં variable_name એ અક્ષરોનો એરે અથવા સ્ટ્રિંગ છે. 'નલ' અક્ષર ન આવે ત્યાં સુધી puts() વિધેય variable_name માં સંગૃહીત તમામ વિગતો મોનિટર પર દર્શાવે છે. એ નોંધ કરો કે દરેક સ્ટ્રિંગના અંતમાં 'નલ' અક્ષર આપેલ હોય છે, પરંતુ puts() વિધેય આ અક્ષરને સ્ક્રીન પર દર્શાવતું નથી. આ અભિગમ સ્પષ્ટ કરવા માટે ઉદાહરણ 12.6નો અભ્યાસ કરો. ઉદાહરણ 12.6 નું કોડ લિસ્ટિંગ આકૃતિ 12.12 માં આપેલ છે તથા આકૃતિ 12.13 તેનું પરિણામ દર્શાવે છે.



```
9-6.c - SciTE
File Edit Search View Tools Options Language Buffers Help
9-6.c
/* Example 6: Program to demonstrate the use of puts() function.*/
#include<stdio.h>
int main()
{
    char state[20];

    printf("Enter your state name:\n");
    gets(state);
    printf("Your state name is ");
    puts(state);
    return 0;
}
//End of Program.
```

આકૃતિ 12.12 : ઉદાહરણ 12.6નું કોડ-લિસ્ટિંગ



```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ ./a.out
Enter your state name:
Gujarat
Your state name is Gujarat
kpp@ubuntu:~$
```

આકૃતિ 12.13 : ઉદાહરણ 12.6નું પરિણામ

આ પ્રોગ્રામ gets() વિધેયનો ઉપયોગ કરી એકથી વધુ અક્ષરો વાંચી તેનો આપેલ ચલમાં સંગ્રહ કરશે. puts() વિધેયના ઉપયોગ દ્વારા ચલમાં આવેલી કિંમતને સ્ક્રીન પર દર્શાવવામાં આવશે.

સુગ્રંથિત પરિણામ (Formatted output)

અત્યાર સુધીમાં ચર્ચવામાં આવેલા આઉટપુટ માટેના વિધેય સુગ્રંથિત (formatted) ન હતા. તેના ચલમાં સંગ્રહવામાં આવેલા કિંમતને માત્ર સ્ક્રીન પર દર્શાવવામાં આવતી હતી. પરંતુ કેટલીકવાર પ્રોગ્રામ પરથી મળતાં પરિણામને એવી

રીતે તૈયાર કરવાની જરૂર પડે છે જે દેખાવમાં આકર્ષક અને સમજવામાં સરળ હોય. પરિણામને જુદી-જુદી સંરચનાઓ સાથે દર્શાવવા માટેની સુવિધા printf() વિધેય દ્વારા પૂરી પાડવામાં આવે છે.

printf()

સ્ક્રીન પર સુગ્રથિત (formatted) પરિણામ દર્શાવવા માટે printf() વિધેયનો ઉપયોગ કરવામાં આવે છે. printf() વિધેયની સામાન્ય વાક્યરચના નીચે આપેલ છે :

printf("control string", var1, var2, ..., varN);

અહીં var1, var2, ..., varN ચલ, અચળ કે પદાવલિ હોઈ શકે છે. પરિણામની ગોઠવણ માટેની સ્પષ્ટતા (output format specification) કંટ્રોલ સ્ટ્રિંગમાં આપવામાં આવે છે. કંટ્રોલ સ્ટ્રિંગમાં નીચે દર્શાવેલ એક અથવા તમામ ઘટકોનો સમાવેશ કરવામાં આવે છે.

- અક્ષરોનો ગણ (સ્ટ્રિંગ) કે જે કંટ્રોલ સ્ટ્રિંગમાં છે તે જ પ્રમાણે મોનિટર પર દર્શાવવાના છે.
- દરેક ચલના ફોર્મેટ સ્પેસિફાયર
- \n (નવી લીટી), \t (ટેબ), \b (બેક સ્પેસ) જેવા એસ્કેપ સિક્વન્સ અક્ષરો

કંટ્રોલ સ્ટ્રિંગમાં આપેલ વિગતોને scanf()ની જેમ જગ્યા આપી છૂટી પાડવામાં આવે છે અને તેની આગળ '%' નિશાની ઉમેરવામાં આવે છે. જે ચલની કિંમત દર્શાવવાની હોય તેનો સમાવેશ printf()ના પછીના ભાગમાં કરવામાં આવે છે. કંટ્રોલ સ્ટ્રિંગમાં આપેલ ફોર્મેટ સ્પેસિફાયર સાથે આ ચલની સંખ્યા, ક્રમ અને પ્રકાર મેળ ખાતા હોવા જોઈએ.

પ્રોગ્રામમાં સ્ક્રીન પર પરિણામ દર્શાવવા માટે આપણે અત્યાર સુધી ઉપયોગમાં લીધેલાં કેટલાક સરળ printf() વિધાનનાં ઉદાહરણ નીચે દર્શાવ્યાં છે :

- printf("Hello World");
- printf("Your age is %d", age);
- printf("Your name is %s", student_name);

જુદા જુદા પ્રકારની ગોઠવણીઓ સાથે વ્યવસ્થિત પરિણામ દર્શાવવા માટે printf() વિધાનની કંટ્રોલ સ્ટ્રિંગમાં ફોર્મેટ સ્પેસિફિકેશન (format specification)નો ઉપયોગ કરવામાં આવે છે. કંટ્રોલ સ્ટ્રિંગનું સામાન્ય સ્વરૂપ નીચે મુજબ છે :

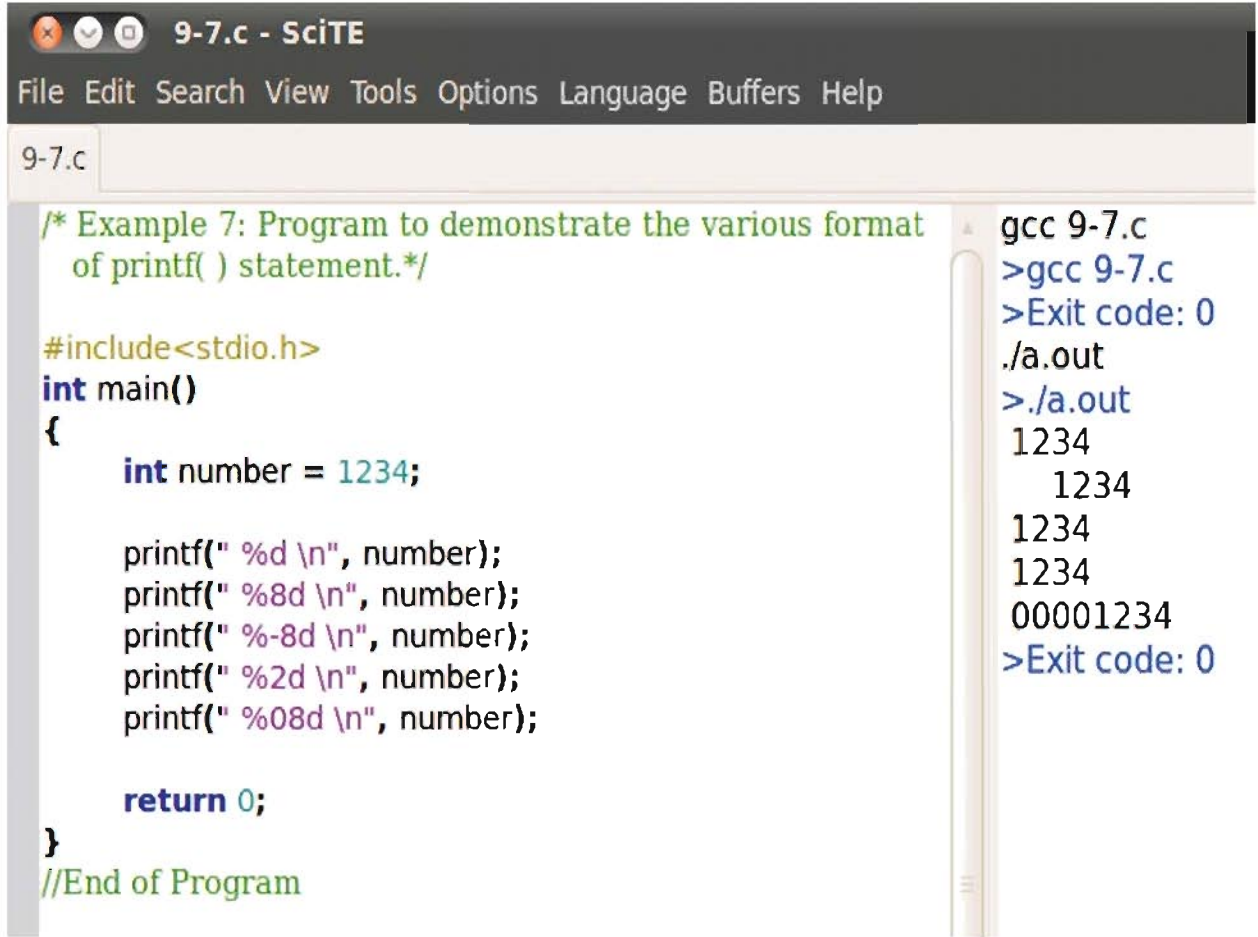
%Lmp

અહીં, 'L' એ કુલ અક્ષરોની સંખ્યાના સ્થાનનો નિર્દેશ કરતો પૂર્ણાંક છે, જેમાં ચલની કિંમત દર્શાવવાની છે. અપૂર્ણ સંખ્યામાં દશાંશ ચિહ્ન પછી દર્શાવવાના અંકો માટે અથવા સ્ટ્રિંગ ચલમાંથી દર્શાવવાના અક્ષરોની સંખ્યા માટે 'm'નો ઉપયોગ કરવામાં આવે છે. l અને mની કિંમતો આપવી વૈકલ્પિક છે. 'p' ચલના પ્રકારનો નિર્દેશ કરે છે. printf() વિધેય સાથે ઉપયોગમાં લઈ શકાય તેવા જુદા-જુદા ડેટાટાઈપની યાદી કોષ્ટક 12.2માં દર્શાવી છે.

ડેટાટાઈપ	સંબંધિત પ્રકાર
દશાંકી સંખ્યા દર્શાવવા માટે	%d
એક અક્ષર દર્શાવવા માટે	%c
ઘાતાંક વગર અપૂર્ણ સંખ્યા દર્શાવવા માટે	%f
ઘાતાંક સાથે અપૂર્ણ સંખ્યા દર્શાવવા માટે	%e
સ્ટ્રિંગ દર્શાવવા માટે	%s
ચિહ્નરહિત (અનસાઈન્ડ) પૂર્ણાંક દર્શાવવા માટે	%u
લોંગ દશાંકી પૂર્ણાંક દર્શાવવા માટે	%ld
ડબલ સંખ્યા દર્શાવવા માટે	%lf
લોંગ ડબલ સંખ્યા દર્શાવવા માટે	%Lf
પૂર્ણાંકને અષ્ટાંકી સ્વરૂપે દર્શાવવા માટે	%o
પૂર્ણાંકને સોળાંકી સ્વરૂપે દર્શાવવા માટે	%x

કોષ્ટક 12.2 : printf() ની કંટ્રોલ સ્ટ્રિંગમાં ઉપયોગી ડેટાટાઈપ

આકૃતિ 12.14માં આપેલ ઉદાહરણ 12.7નું કોડ લિસ્ટિંગ સમજવાનો પ્રયત્ન કરીએ, જે printf() વિધાનમાં વિવિધ સ્વરૂપો દર્શાવે છે.



The screenshot shows a code editor window titled "9-7.c - SciTE". The menu bar includes File, Edit, Search, View, Tools, Options, Language, Buffers, and Help. The file name "9-7.c" is in the title bar. The code in the editor is as follows:

```

/* Example 7: Program to demonstrate the various format
of printf( ) statement.*/

#include<stdio.h>
int main()
{
    int number = 1234;

    printf(" %d \n", number);
    printf(" %8d \n", number);
    printf(" %-8d \n", number);
    printf(" %2d \n", number);
    printf(" %08d \n", number);

    return 0;
}
//End of Program

```

On the right side, the output of the program is shown, generated by gcc 9-7.c. The output is:

```

gcc 9-7.c
>gcc 9-7.c
>Exit code: 0
./a.out
>./a.out
1234
    1234
1234
1234
00001234
>Exit code: 0

```

આકૃતિ 12.14 : ઉદાહરણ 12.7નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી (Explanation)

પ્રથમ પરિણામમાં દર્શાવ્યું છે તે મુજબ પૂર્વનિર્ધારિત રીતે printf() વિધાન પરિણામ દર્શાવવા માટે ડાબી બાજુની ગોઠવણ(left align)નો ઉપયોગ કરે છે. બીજું printf() વિધાન જમણી બાજુની ગોઠવણ (right justification) સાથે 8 અક્ષરોની જગ્યાનો ઉપયોગ કરે છે. ત્રીજું printf() વિધાન દર્શાવે છે કે % નિશાની પછી ઋણ નિશાની (-) ઉમેરવાથી પરિણામને ડાબી બાજુ ગોઠવવું શક્ય છે. ચોથા printf() વિધાનમાં %2d લખવાથી કોઈ અસર મેળવી શકાતી નથી. કારણ કે, આપવામાં આવેલ સંખ્યા ચલના અક્ષરોની કુલ સંખ્યા કરતાં ઓછી છે. અંતિમ printf() વિધાન દર્શાવે છે કે ચલની મૂળ કિંમત સાથે વધારાના અક્ષરો (શૂન્ય) દર્શાવવાનું પણ શક્ય છે.

અગાઉના ઉદાહરણમાં આપણે જોયું કે પૂર્ણાંક ચલને જુદા જુદા પ્રકારની સંરચનાઓ સાથે કેવી રીતે દર્શાવી શકાય. હવે, વિવિધ સંરચનાઓ દ્વારા અપૂર્ણાંક સંખ્યાને કેવી રીતે દર્શાવી શકાય તે જોઈએ. સામાન્ય રચના %l.mmp યાદ કરો જેમાં m એ દશાંશ પછી દર્શાવવાના અંકોનો નિર્દેશ કરે છે. જ્યારે અપૂર્ણાંક સંખ્યા દર્શાવવામાં આવે છે ત્યારે l સ્તંભની પહોળાઈમાં m દશાંશ જગ્યાઓ સાથે જમણી બાજુ ગોઠવાયેલ અંકો રજૂ કરવામાં આવે છે. જો mનો ઉપયોગ કર્યો ન હોય તો દશાંશચિહ્ન પછી પૂર્વનિર્ધારિત 6 અંક દર્શાવવામાં આવે છે. f1 ચલમાં આપેલી કિંમત 123.456 વિવિધ સંરચનાઓનો ઉપયોગ કરી printf() વિધાનની મદદથી કેવી રીતે રજૂ કરી શકાય તે આકૃતિ 12.15માં દર્શાવ્યું છે.

Statement	Output
<code>printf(" %f", f1);</code>	1 2 3 . 4 5 6 0 0 0
<code>printf(" %8.3f", f1);</code>	1 2 3 . 4 5 6
<code>printf(" %8.1f", f1);</code>	1 2 3 . 5
<code>printf(" %08.1f", f1);</code>	0 0 0 1 2 3 . 5
<code>printf(" %-8.1f", f1);</code>	1 2 3 . 5
<code>printf(" %10.3e", f1);</code>	1 . 2 3 5 e + 0 2
<code>printf(" %10.4e", f1);</code>	1 . 2 3 4 6 e + 0 2

આકૃતિ 12.15 : વિવિધ સંરચનાઓ દ્વારા અપૂર્ણાંક સંખ્યાની રજૂઆત

`printf()` વિધાનનો ઉપયોગ કરી અક્ષર અને સ્ટ્રિંગની સંરચના પણ કરી શકાય છે. આ માટેનું ફોર્મેટ સ્પેસિફિકેશન અપૂર્ણાંક સંખ્યા જેવું જ છે. *f*ની કિંમતનો ઉપયોગ ફિલ્ડની પહોળાઈ દર્શાવવા માટે કરવામાં આવે છે, જ્યારે દર્શાવવામાં આવનાર અક્ષરોની સંખ્યાનો નિર્દેશ *m*ની કિંમત દ્વારા કરવામાં આવે છે. જો *f*ની કિંમતનો ઉપયોગ કરવામાં ન આવે તો સ્ટ્રિંગને ડાબી બાજુ ગોઠવવામાં આવે છે તથા *f*નો ઉપયોગ સ્ટ્રિંગને સ્ક્રીન પર જમણી બાજુ ગોઠવે છે. પૂર્ણાંક સંખ્યાના ફોર્મેટ સ્પેસિફિકેશન મુજબ ઋણ નિશાની ઉમેરવાથી સ્ટ્રિંગને ડાબી બાજુ ગોઠવી શકાય છે.

નીચેના `printf()` વિધાનનો ઉપયોગ કરી સ્ટ્રિંગ માટેની સંરચનાઓ ખ્યાલ વધુ સ્પષ્ટ કરીએ. અહીં `str` નામના સ્ટ્રિંગ ચલ (અક્ષરોના એરે)માં "GUJARAT INDIA" કિંમતનો સંગ્રહ કરવામાં આવ્યો છે. આકૃતિ 12.16 આ વિવિધ સંરચનાઓ દર્શાવે છે.

Statement	Output
<code>printf(" %s", str);</code>	G U J A R A T I N D I A
<code>printf(" %8s", str);</code>	G U J A R A T I N D I A
<code>printf(" %16s", str);</code>	G U J A R A T I N D I A
<code>printf(" %16.7s", str);</code>	G U J A R A T
<code>printf(" %.7s", str);</code>	G U J A R A T
<code>printf(" %-16.9s", str);</code>	G U J A R A T I
<code>printf(" %16.15s", str);</code>	G U J A R A T I N D I A

આકૃતિ 12.16 : વિવિધ સંરચનાઓ દ્વારા અક્ષરોની રજૂઆત

અંતિમ `printf()` વિધાનમાં આપેલ *m*ની કિંમત 15 `str` ચલમાં સંગ્રહવામાં આવેલા સ્ટ્રિંગની લંબાઈ કરતાં વધુ હોવાથી `str` ચલના તમામ અક્ષરો જમણી બાજુ દર્શાવવામાં આવશે.

સારાંશ

સુગ્રાહિત નિવેશ અને નિર્ગમ આંતરસ્થાપિત વિધેય (formatted input and output inbuilt function)નો પ્રોગ્રામમાં કેવી રીતે ઉપયોગ થઈ શકે તે માટેનો અભ્યાસ આપણે આ પ્રકરણમાં કર્યો. પરિણામને સંરચના (formation) વગર પણ દર્શાવી શકાય છે, પરંતુ વિશિષ્ટ સંરચના દ્વારા રજૂ કરેલું પરિણામ ઉપયોગકર્તાને વધુ સુવાચ્ય અને આકર્ષક લાગે છે. પ્રોગ્રામને આકર્ષક બનાવવા આ વિધેયનો ઉપયોગ કેવી રીતે કરવો તેનો આધાર પ્રોગ્રામર પર છે.

સ્વાધ્યાય

1. `scanf()` વિધેયની કન્ટ્રોલ સ્ટ્રિંગ `printf()` વિધેયની કન્ટ્રોલ સ્ટ્રિંગ કરતા કેવી રીતે અલગ પડે છે ?
2. પ્રોગ્રામમાં સંરચના કરી શકાય તે પ્રકારના આઉટપુટ વિધેય(formatted output function)ના ઉપયોગની ચર્ચા કરો.
3. પ્રોગ્રામમાં સંરચના કરી શકાય તે પ્રકારના ઇનપુટ વિધેય(formatted input function)ના ઉપયોગની ચર્ચા કરો.
4. યોગ્ય ઉદાહરણનો ઉપયોગ કરી `%[characters]` અને `%[^characters]`ની મદદથી મર્યાદિત ઇનપુટનો અભિગમ સમજાવો.
5. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો :
 - (a) એક સમયે એકથી વધુ અક્ષરો ઉમેરવા `getchar()` વિધેયનો ઉપયોગ કરી શકાય.
 - (b) `gets()` વિધેયના ઉપયોગ દ્વારા એક અક્ષર વાંચી શકાય છે.
 - (c) એક જ `printf()` વિધાનની મદદથી એકથી વધુ લીટીઓમાં પરિણામ દર્શાવી શકાય છે.
 - (d) `%.10s` ફોર્મેટ સ્પેસિફાયર સ્ટ્રિંગના પ્રથમ દસ અક્ષરો દર્શાવશે.
 - (e) એકથી વધુ ક્રિમતો મેળવવા એક જ `scanf()` વિધાનનો ઉપયોગ કરી શકાય છે.
 - (f) અપૂર્ણ સંખ્યાઓને પૂર્વનિર્ધારિત રીતે છ દશાંશ સાથે દર્શાવવામાં આવે છે.
6. માગ્યા મુજબ કરો :
 - (1) નીચેનાં વિધાનોમાં કોઈ ક્ષતિ હોય તો જણાવો :
 - (a) `scanf("%d %c", &number, city_code);`
 - (b) `scanf("%d %d %s" \n, &marks1, &marks2, city_name);`
 - (c) `scanf("%d %f %c %s", &num1, &price, item_code);`
 - (2) નીચેનાં વિધાનોનું પરિણામ જણાવો :
 - (a) `printf("%d %c %f", 101, 'X', 20.20);`
 - (b) `printf("%3d %8.2f ", 12345, 222.123);`
 - (c) `printf("%.5s", "Hello World");`
 - (d) `printf("%15.8s", "Hello Student");`
 - (e) `printf("%.5s", "Hello World");`

(3) નીચેના પ્રોગ્રામનું પરિણામ લખો :

```
#include<stdio.h>

int main( )
{
    float result = 98.12345;

    printf("Your result is %f \n", result);
    printf("Your result is %.2f \n", result);
    printf("Your result is %5.3f \n", result);
    return 0;
}
```

(4) નીચેના પ્રોગ્રામમાં કોઈ ક્ષતિ હોય તો તેને દૂર કરી પરિણામ લખો :

```
#include<std.h>

int main( )
{
    int sum = 1234;
    float price = = 550.50;
    char city[20] = "Ahmedabad";

    printf("The sum is %d....", price);
    printf("The price is %s...", price);
    printf("The city name is %.3s...", city);
    return 0;
}
```

7. આપેલ વિકલ્પોમાંથી સાચો વિકલ્પ પસંદ કરો :

(1) સી ભાષામાં I/O પ્રક્રિયા માટે નીચેનામાંથી શેનો ઉપયોગ કરવામાં આવે છે ?

(a) મોનિટર (b) કી-બોર્ડ (c) માઉસ (d) આપેલ તમામ

(2) નીચેનામાંથી કયા સાધન દ્વારા પ્રોગ્રામમાં ઈનપુટ શક્ય છે ?

(a) કી-બોર્ડ (b) સ્પીકર (c) પ્રિન્ટર (d) મોનિટર

(3) નીચેનામાંથી કયું વિધેય ઇનપુટ માટે નથી ?

- (a) getch() (b) getch() (c) puts() (d) gets()

(4) printf("%4s", "Krusha");નું પરિણામ શું મળશે ?

- (a) Krus (b) usha (c) Krusha (d) Krush

(5) સ્ટ્રિંગ સ્વીકારવા માટે નીચેનામાંથી કયું વિધેય વધુ યોગ્ય છે ?

- (a) getch() (b) gets() (c) getchar() (d)getc()

પ્રાયોગિક સ્વાધ્યાય

1. ઉપયોગકર્તા પાસેથી "Hello Gujarat" સ્ટ્રિંગ મેળવી, નીચે આપેલ જુદા જુદા સ્વરૂપે દર્શાવો :

- (a) Hello (b) Gujarat (c) Hello Gujarat (d) Hello G

2. ઉપયોગકર્તાના ઇનપુટને આધારે ઘડિયો (Multiplication table) દર્શાવે તેવો સી પ્રોગ્રામ બનાવો. પરિણામને વિવિધ સંરચનાઓ સાથે દર્શાવવા આઉટપુટ ફોર્મેટિંગના જુદા જુદા વિકલ્પોનો ઉપયોગ કરો.

3. ઉપયોગકર્તા પાસેથી માત્ર કેપિટલ અક્ષરો સ્વીકારી શકાય તેવો પ્રોગ્રામ બનાવો. (સી ભાષામાં ઉપલબ્ધ મર્યાદિત ઇનપુટના અભિગમનો ઉપયોગ કરો.)

4. ઉપયોગકર્તા પાસેથી થોડા પૂર્ણાંકો મેળવવા માટેનો પ્રોગ્રામ બનાવો. તમામ પૂર્ણાંકોને સ્ક્રીન પર જમણી બાજુ દર્શાવો.

5. નીચેના અંકો મેળવવા માટેનો સી પ્રોગ્રામ બનાવો :

123.45 34.56 - 7878.123 - 0.965

યોગ્ય રેખાપ્રકાર ધરાવતા ચલમાં આ અંકોનો સંગ્રહ કરો. તમામ અંકોને તેની પાસેના પૂર્ણાંકોમાં ફેરવીને દર્શાવો.

6. A અને Bની કિંમત મેળવવા માટેનો સી પ્રોગ્રામ લખો. ત્યાર પછી નીચેની પદાવલિઓનું પરિણામ એક જ લીટીમાં દર્શાવો :

- (a) $(A + B) / (A - B)$ (b) $(A + B) (A - B)$ (c) $(A * A) (B + B)$



નિર્ણય માળખાં

અત્યાર સુધી રજૂ કરવામાં આવેલા મોટાભાગના તમામ સી પ્રોગ્રામ તેના અમલ વખતે એક ક્રમબદ્ધ માળખાને અનુસરે છે. પ્રોગ્રામમાં લખવામાં આવેલા ક્રમ પ્રમાણે એક પછી એક તમામ સૂચનાઓનો અમલ કરવામાં આવે છે.

રોજિંદા જીવનમાં સી પ્રોગ્રામના અમલ દરમિયાન આપણને કેટલીક શરતોને આધારે અમુક જ વિધાનોનો અમલ કરવાની જરૂર પડે એમ પણ બને. આ હેતુ પાર પાડવા માટે સૂચનાઓના ક્રમનો પ્રવાહ બદલવો જરૂરી બને છે. પ્રોગ્રામમાં સૂચનાઓના ક્રમનો પ્રવાહ બદલી શકાય તે માટે સી ભાષા વિશિષ્ટ પ્રકારનાં વિધાનોની સુવિધા પૂરી પાડે છે. આ વિધાનોને નિર્ણય માળખાંનાં વિધાનો તરીકે ઓળખવામાં આવે છે.

સી ભાષામાં નિર્ણય માળખાંની જરૂરિયાત (Need for decision structure in C)

કોઈ શરતના પરિણામને આધારે પ્રોગ્રામના એક વિભાગ પરથી અન્ય વિભાગ પર જવા માટે નિર્ણય માળખાં મદદરૂપ બને છે. કેટલીકવાર નિર્ણય માળખાંનાં વિધાનોને પસંદગી માળખાંનાં વિધાનો (selective structure statements), શાખાકીય વિધાનો (branching statements) અથવા નિર્ણય લેનાર વિધાનો (decision making statements) તરીકે પણ ઓળખવામાં આવે છે. અમલીકરણના પ્રવાહને નિયંત્રિત કરતાં હોવાથી આ વિધાનોને નિયંત્રણ વિધાનો (control statements) પણ કહેવામાં આવે છે.

સી ભાષા મૂળભૂત રીતે બે પ્રકારનાં નિર્ણય માળખાં માટેનાં વિધાનો પૂરાં પાડે છે : **if** અને **switch**. પ્રોગ્રામમાં સૂચનાઓનો પ્રવાહ **if** અને **switch** વિધાનોની મદદથી કેવી રીતે બદલી શકાય તેનો આપણે આ પ્રકરણમાં અભ્યાસ કરીશું.

if વિધાન (The if statement)

નિર્ણય માળખા માટેનું **if** એક એવું સક્ષમ વિધાન છે, જે સૂચનાઓના અમલને નિયંત્રિત કરવા માટે ઉપયોગમાં લઈ શકાય છે. **if** વિધાનનો ઉપયોગ નીચે જણાવેલ વિવિધ પ્રકારે કરી શકાય છે :

- સરળ if વિધાન (simple if statement)
- if-else વિધાન (if-else statement)
- નેસ્ટેડ if વિધાન (nested if statement)
- else-if લેડર વિધાન (else-if ladder statement)

સરળ if વિધાન (Simple if statement)

નિર્ણય માળખાનું સરળતમ સ્વરૂપ if વિધાન છે. નિર્ણયો લેવા માટે તથા પ્રોગ્રામના અમલીકરણનો પ્રવાહ બદલવા માટે આ વિધાનનો વારંવાર ઉપયોગ કરવામાં આવે છે.

ઉદાહરણ 13.1 : આપણે એવો પ્રોગ્રામ સમજવાનો પ્રયત્ન કરીએ, જે ઉપયોગકર્તા પાસેથી કોઈ એક વિષયના ગુણ મેળવે અને જો તે 80થી વધુ હોય તો અભિનંદનનો સંદેશ દર્શાવે. ઉદાહરણ 13.1 માટેનું કોડ લિસ્ટિંગ આકૃતિ 13.1માં તથા તેનું પરિણામ આકૃતિ 13.2માં દર્શાવ્યું છે.

```
10-1.c - SciTE
File Edit Search View Tools Options Language Buffers Help
10-1.c
/* Example 1: Program to illustrate use of simple if statement */
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int marks;                // declare a variable
    system("clear");          // clears the screen
    printf("Enter your marks:"); // message to the user
    scanf("%d", &marks);      // read a number from user

    if ( marks > 80 )          // check whether marks greater than 80 or not
        printf("Congratulations...\n"); // display message only when marks > 80

    printf("Have a Nice Time...\n");
    return 0;
}

/* End of Program */
```

આકૃતિ 13.1 : ઉદાહરણ 13.1 માટેનું કોડ-વિસ્ટિંગ

```
kpp@ubuntu: ~
File Edit View Terminal Help
Enter your marks:85
Congratulations...
Have a Nice Time...
kpp@ubuntu:~$
```

આકૃતિ 13.2 : ઉદાહરણ 13.1નું પરિણામ

સમજૂતી

main() વિધેયમાં આવેલ પ્રથમ વિધાન marks નામના ચલને ઘોષિત કરે છે. બીજું વિધાન સ્ક્રીન પરથી લખાણ દૂર કરે છે. ઉપયોગકર્તાને સંદેશ દર્શાવવા માટે printf વિધાનનો ઉપયોગ કર્યો છે. scanf વિધાનના ઉપયોગ દ્વારા ઉપયોગકર્તા પાસેથી કિંમત મેળવી તેનો marks નામના ચલમાં સંગ્રહ કરવામાં આવે છે. if વિધાન શરત ચકાસે છે : દાખલ કરવામાં આવેલ ગુણ 80થી વધુ છે? જો ટેસ્ટ એક્સપ્રેશન (marks > 80)નું પરિણામ True (સાચું) મળશે તો "Congratulations..." સંદેશ ધરાવતા printf વિધાનનો અમલ કરવામાં આવશે. જો શારતિક વિધાન (marks > 80)નું પરિણામ false (ખોટું) મળશે તો "Congratulations..." સંદેશ દર્શાવતા printf વિધાનનું અમલીકરણ મોકૂફ રાખવામાં આવશે. ત્યાર પછીનું printf વિધાન "Have a nice time..." સંદેશ હંમેશા દર્શાવશે.

સરળ if વિધાનની વાક્યરચના (Syntax of simple if statement)

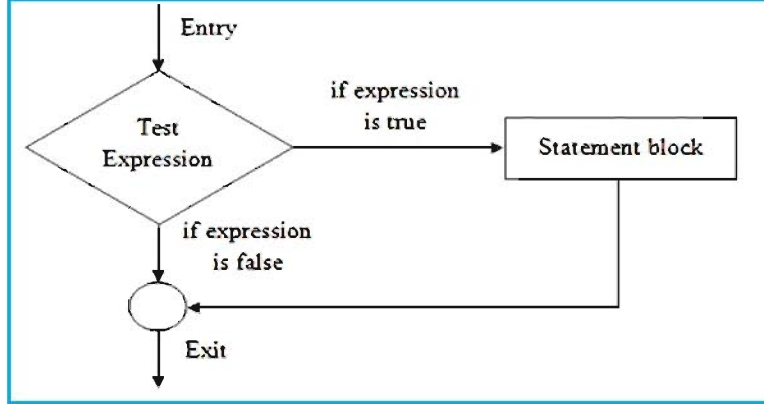
નીચેની વાક્યરચના દ્વારા સરળ if વિધાનનો ઉપયોગ કરી શકાય :

```
if (test expression)
{
    statement-block;
}
```

અહીં test expression એ સી ભાષામાં માન્ય એવી કોઈ પણ પદાવલિ છે. ટેસ્ટ એક્સપ્રેશનમાં તાર્કિક પદાવલિ (logical expression)નો ઉપયોગ કરવામાં આવે છે. statement-block તરીકે સી ભાષામાં માન્ય એવા કોઈ પણ એક કે વધુ વિધાનો હોઈ શકે છે. અહીં એ ધ્યાન આપો કે ટેસ્ટ એક્સપ્રેશનની લીટીને અર્ધવિરામ દ્વારા પૂર્ણ કરવામાં આવતી નથી.

આ વાક્યરચના દર્શાવે છે કે જો ટેસ્ટ એક્સપ્રેશન True હશે તો સ્ટેટમેન્ટ બ્લોકનો અમલ કરવામાં આવશે અને જો ટેસ્ટ એક્સપ્રેશન False હશે તો સ્ટેટમેન્ટ બ્લોકને અવગણીને પ્રોગ્રામમાં તેના પછી આવેલ સૂચનાઓનો અમલ કરવામાં આવશે.

સી પ્રોગ્રામિંગ ભાષા શૂન્યેતર અને ખાલી ન હોય તેવી (non-null) કિંમતોને True તરીકે મૂલવે છે તથા શૂન્ય કે ખાલી (null) કિંમતોને False તરીકે મૂલવે છે. આકૃતિ 13.3 સરળાંક વિધાનનો ફ્લો-ચાર્ટ દર્શાવે છે.



આકૃતિ 13.3 : સરળ if વિધાનનો ફ્લો-ચાર્ટ

if...else વિધાન (The if...else statement)

સરળ if વિધાનમાં માત્ર એક જ સ્ટેટમેન્ટ બ્લોક હોય છે, અને જો ટેસ્ટ એક્સપ્રેશન સાચું હોય તો જ તેનો અમલ કરવામાં આવે છે. ટેસ્ટ એક્સપ્રેશન ખોટું હોય ત્યારે સરળ if વિધાન કોઈ કાર્યનો અમલ કરતું નથી. હવે, ટેસ્ટ એક્સપ્રેશનનો જવાબ False આવે ત્યારે જો કોઈ ક્રિયા અમલમાં મૂકવી હોય તો? ઉદાહરણ તરીકે, જો વિદ્યાર્થીના ગુણ 40થી વધુ હોય તો આપણે "Student Passed" સંદેશ દર્શાવવા માંગીએ છીએ અન્યથા "Student Failed" સંદેશ દર્શાવવામાં આવવો જોઈએ. આવા કિસ્સામાં if...else માળખું ઉપયોગી બને છે.

ઉદાહરણ 13.2 : આપણે એક એવો પ્રોગ્રામ સમજવાનો પ્રયત્ન કરીએ જે એક વિષયના ગુણ મેળવી, જો ગુણ 40થી વધુ હોય તો "Congratulations...You are passed" સંદેશ દર્શાવે, અન્યથા "Better Luck Next Time...You are failed" સંદેશ દર્શાવે. આકૃતિ 13.4માં ઉદાહરણ 13.2નું કોડ લિસ્ટિંગ આપેલ છે તથા તેનાં બે જુદાં જુદાં પરિણામ આકૃતિ 13.5માં દર્શાવ્યાં છે.

```

10-2.c - SciTE
File Edit Search View Tools Options Language Buffers Help

10-2.c
/* Example 2: Program to illustrate use of if...else statement. */
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int marks;                // declare a variable
    system("clear");          // clears the screen
    printf("Enter your marks:"); // message to the user
    scanf("%d", &marks);      // read a number from user

    if ( marks > 40 )          // check whether marks greater than 40
    {
        printf("Congratulations...you are passed.\n"); // Condition true
    }
    else
    {
        printf("Better Luck Next Time...you are failed.\n"); // Condition false
    }

    printf("Have a Nice Time...\n");
    return 0;
}
/* End of Program */
  
```

આકૃતિ 13.4 : ઉદાહરણ 13.2નું કોડ-લિસ્ટિંગ


```
kpp@ubuntu: ~
File Edit View Terminal Help
Enter your marks:80
Congratulations...you are passed.
Have a Nice Time...
kpp@ubuntu:~$
```

```
kpp@ubuntu: ~
File Edit View Terminal Help
Enter your marks:35
Better Luck Next Time...you are failed.
Have a Nice Time...
kpp@ubuntu:~$
```

આકૃતિ 13.5 : ઉદાહરણ 13.2નું પરિણામ

સમજૂતી

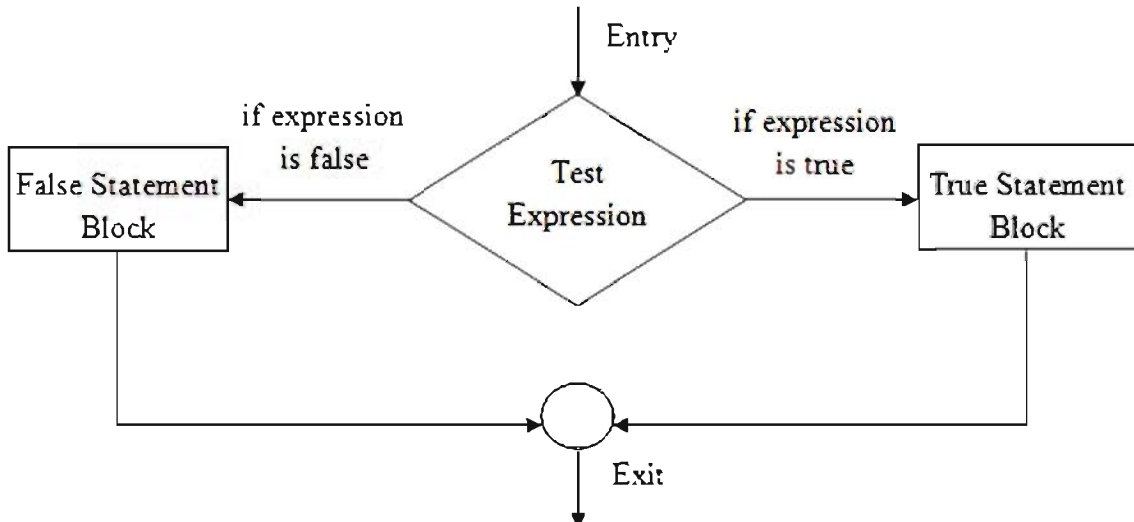
પ્રોગ્રામનું પ્રથમ વિધાન marks નામનો ચલ ઘોષિત કરે છે. બીજું વિધાન સ્ક્રીન પરથી તમામ લખાણ દૂર કરે છે. ત્રીજા અને ચોથા વિધાન દ્વારા જરૂરી સંદેશ દર્શાવી ઉપયોગકર્તા પાસેથી marks ચલની કિંમત મેળવવામાં આવે છે. આપેલ ગુણ 40થી વધુ છે કે નહીં તે ચોથા વિધાન દ્વારા ચકાસવામાં આવે છે. જો ઉપયોગકર્તા 40થી વધુ ગુણ ઉમેરશે તો "Congratulations...You are passed" સંદેશ દર્શાવવામાં આવશે, અન્યથા "Better Luck Next Time...You are failed" સંદેશ દર્શાવાશે. ત્યાર પછી આવેલું printf વિધાન હંમેશા "Have a nice time..." સંદેશ હંમેશાં રજૂ કરશે.

if...else વિધાનની વાક્યરચના (Syntax of if...else statement)

if...else વિધાનની વાક્યરચના નીચે આપેલ છે :

```
if (test expression)
{
    True statement-block;
}
else
{
    False statement-block;
}
```

અહીં test expression એ સી ભાષાની કોઈ પણ યોગ્ય પદાવલિ હોઈ શકે છે. જ્યારે test expressionનો જવાબ True મળશે ત્યારે નિયંત્રણનો પ્રવાહ True statement-block તરફ જશે. એથી વિપરીત જો ટેસ્ટ એક્સપ્રેશનનો જવાબ False મળશે તો, નિયંત્રણ પ્રવાહ False statement-block તરફ જશે. બંને કિસ્સામાં if વિધાનનો અમલ કર્યા બાદ નિયંત્રણને પ્રોગ્રામમાં આવેલા તે પછીના વિધાન તરફ મોકલવામાં આવશે. આ અભિગમ આકૃતિ 13.6માં if...else વિધાનના ફ્લોચાર્ટ દ્વારા સમજાવવામાં આવ્યો છે.



આકૃતિ 13.6 : if...else વિધાનનો ફ્લોચાર્ટ

નેસ્ટેડ if...else વિધાન (Nested if...else statement)

પ્રોગ્રામમાં ક્યારેક એકથી વધુ નિર્ણયોની શ્રેણીનો અમલ કરાવવાની પણ જરૂર પડે છે. આ હેતુ પાર પાડવા માટે if...else વિધાનની શ્રેણીનો 'નેસ્ટેડ' સ્વરૂપે ઉપયોગ કરી શકાય છે. નીચેની વાક્યરચના જુઓ. જેમાં એક if...else વિધાનનો ઉપયોગ અન્ય if...else વિધાનના બ્લૉકની અંદર કરવામાં આવ્યો છે.

```
if (test expression-1)
{
    if(test expression-2)
    {
        Statement block-1;
    }
    else
    {
        Statement block-2;
    }
}
else
{
    Statement block-3;
}
```

નેસ્ટેડ if વિધાનમાં અમલીકરણની તક નીચે દર્શાવેલ છે :

- જો ટેસ્ટ એક્સપ્રેશન-1 સાચું હશે તો ટેસ્ટ એક્સપ્રેશન-2 ચકાસવામાં આવશે.
- જો ટેસ્ટ એક્સપ્રેશન-2 પણ સાચું હશે તો સ્ટેટમેન્ટ બ્લૉક-1નો અન્યથા સ્ટેટમેન્ટ બ્લૉક-2નો અમલ કરવામાં આવશે.
- જો ટેસ્ટ એક્સપ્રેશન-1 ખોટું હોય તો સ્ટેટમેન્ટ બ્લૉક-3નો અમલ કરવામાં આવશે. અહીં એ નોંધવું જરૂરી છે કે સ્ટેટમેન્ટ બ્લૉક-3ના ભાગ તરીકે વધુ if...else વિધાનોનો ઉપયોગ કરી શકાય છે.

નોંધ : ઉપર જણાવેલ બંધારણ માત્ર ઉદાહરણ માટે છે. ઉપયોગકર્તા પોતાની જરૂરિયાત મુજબ તેને બદલી શકે છે. ઉદાહરણ તરીકે, નીચે જણાવેલ બંધારણ પણ યોગ્ય છે :

```
if (test expression-1)
{
    Statement block-1;
}
else
{
    if(test expression-2)
    {
        Statement block-2;
    }
    else
    {
        Statement block-3;
    }
}
```

ઉદાહરણ 13.3 : ઉપયોગકર્તા પાસેથી ત્રણ સંખ્યાઓ મેળવીને નેસ્ટેડ if વિધાનની મદદથી તેમાંથી સૌથી મોટી સંખ્યા દર્શાવવા માટેના સી પ્રોગ્રામને સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 13.7માં ઉદાહરણ 13.3નું કોડ-લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 13.8 તેનું પરિણામ દર્શાવે છે.

```
10-3.c - SciTE
File Edit Search View Tools Options Language Buffers Help

10-3.c
/* Example 3: Program to demonstrate the use of nested if statement. */
#include<stdio.h>

int main()
{
    int number1, number2, number3;    //declares three variables
    printf("Enter three numbers:");    // message to the user
    scanf("%d %d %d", &number1, &number2, &number3); //read 3 numbers

    if ( number1 > number2 )           // test condition 1
    {
        if ( number1 > number3 )       // test condition 2
            printf("The largest number is %d \n", number1);
        else
            printf("The largest number is %d \n", number3);
    }
    else
    {
        if ( number3 > number2 )       // test condition 3
            printf("The largest number is %d \n", number3);
        else
            printf("The largest number is %d \n", number2);
    }
    return 0;
}
/* End of Program.*/
```

આકૃતિ 13.7 : ઉદાહરણ 13.3નું કોડ-લિસ્ટિંગ

```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 10-3.c
kpp@ubuntu:~$ ./a.out
Enter three numbers:4 6 8
The largest number is 8
kpp@ubuntu:~$
```

આકૃતિ 13.8 : ઉદાહરણ 13.3નું પરિણામ

સમજૂતી

main() વિધેયની અંદર આવેલું પ્રથમ વિધાન ત્રણ ચલ ઘોષિત કરશે. બીજું વિધાન ઉપયોગકર્તા સમક્ષ સંદેશ દર્શાવશે. ત્રીજું વિધાન ઉપયોગકર્તાને ત્રણ કિંમતો ઉમેરવાની પરવાનગી આપશે, જેને number1, number2 અને number3 નામના ચલમાં સંગૃહીત કરવામાં આવશે. ચોથા વિધાન (ટેસ્ટ કન્ડિશન-1) દ્વારા ચકાસણી કરવામાં આવશે કે number1ની કિંમત number2થી વધુ છે કે નહીં. જો ટેસ્ટ કન્ડિશન-1 સાચી હશે તો અંદરના if વિધાન(ટેસ્ટ કન્ડિશન-2)નો અમલ કરી number1 > number3 છે કે નહીં તે ચકાસવામાં આવશે. ટેસ્ટ કન્ડિશન-2ના સાચા કે ખોટા પરિણામને આધારે અંદરનું if વિધાન ઉપયોગકર્તા સમક્ષ સંદેશ રજૂ કરશે. પરંતુ જો ચોથું વિધાન (ટેસ્ટ કન્ડિશન-1) ખોટું પડશે તો બહારના if વિધાનનો else વિભાગ અમલમાં મૂકાશે. અહીં, ટેસ્ટ કન્ડિશન-3 ચકાસશે કે number3 > number2 છે કે નહીં, અને તેના સાચા કે ખોટા પરિણામને આધારે ઉપયોગકર્તા સમક્ષ સંદેશ દર્શાવવામાં આવશે.

else...if લેડર વિધાન (The else...if ladder statement)

એકથી વધારે નિર્ણયોનું મૂલ્યાંકન કરવા માટે એક કરતાં વધુ if વિધાનોની જરૂર પડે છે. આ માટે એક અન્ય બંધારણ પણ ઉપલબ્ધ છે. જ્યારે અન્ય if વિધાનોનો નેસ્ટેડ ifના else (false) બ્લોકમાં ઉપયોગ કરવામાં આવે ત્યારે તે else...if લેડર વિધાન રજૂ કરે છે.

આપેલ વિષયના ગુણ અનુસાર વિદ્યાર્થીની વર્ગ / શ્રેણી આપવા માટેના પ્રોગ્રામમાં else...if લેડરનો ઉપયોગ કરીએ. શ્રેણી આપવા માટેના પ્રોગ્રામમાં આપણે કોષ્ટક 13.1માં આવેલ નિયમોનો ઉપયોગ કરીશું.

ગુણ-વિસ્તાર	વર્ગ / શ્રેણી
70 – 100	First Class with Distinction
60 – 69	First Class
50 – 59	Second Class
40 – 49	Pass Class
0 – 39	Fail

કોષ્ટક 13.1 : નમૂનારૂપ વર્ગના નિયમો

ઉદાહરણ 13.4 : ઉપયોગકર્તા પાસેથી ગુણ મેળવી તેના પરથી વર્ગ / શ્રેણી શોધી આપે તેવા એક C પ્રોગ્રામને સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 13.9 ઉદાહરણ 13.4નું કોડ લિસ્ટિંગ દર્શાવે છે તથા આકૃતિ 13.10માં તેનાં બે પરિણામ દર્શાવ્યાં છે.

```
10-4.c - SciTE
File Edit Search View Tools Options Language Buffers Help

10-4.c
/* Example 4: Program to illustrate use of else...if ladder statement. */
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int marks;                                // declare a variable
    system("clear");                          //clears the screen
    printf("Enter your marks:");              // message to the user
    scanf("%d", &marks);                      // read a number from user

    if ( marks > 69 )                          // check condition 1
        printf("First Class with Distinction \n");
    else if ( marks > 59 )                     // check condition 2
        printf("First Class \n");
    else if ( marks > 49 )                     // check condition 3
        printf("Second Class \n");
    else if ( marks > 39 )                     // check condition 4
        printf("Pass Class \n");
    else                                      // default else statement
        printf("Fail \n");

    return 0;
}
/* End of Program */
```

આકૃતિ 13.9 : ઉદાહરણ 13.4નું કોડ લિસ્ટિંગ


```
kpp@ubuntu: ~  
File Edit View Terminal Help  
Enter your marks:80  
First Class with Distinction  
kpp@ubuntu:~$
```

```
kpp@ubuntu: ~  
File Edit View Terminal Help  
Enter your marks:55  
Second Class  
kpp@ubuntu:~$
```

આકૃતિ 13.10 : ઉદાહરણ 13.4નું પરિણામ

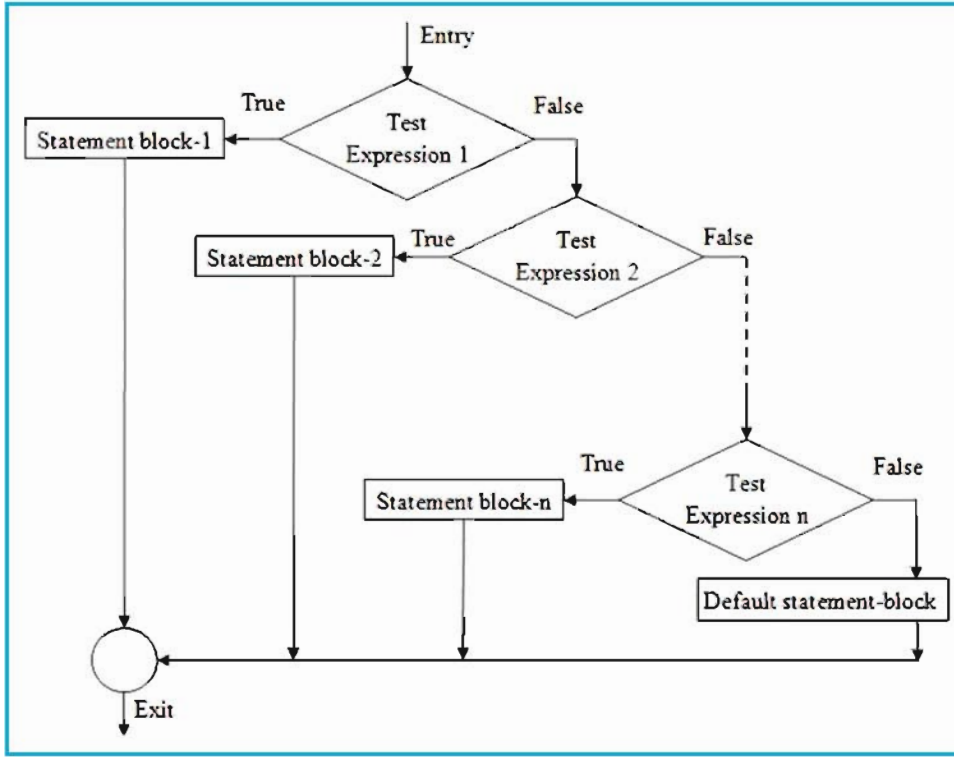
સમજૂતી

main() વિષય પછીનું પ્રથમ, બીજું, ત્રીજું અને ચોથું વિધાન અનુક્રમે ચલ ઘોષિત કરે છે, સ્ક્રીન ચોખ્ખી કરે છે, સંદેશ દર્શાવે છે અને ગુણ સ્વીકારે છે. પાંચમું વિધાન પ્રોગ્રામની પ્રથમ શરત marks > 69 તપાસે છે. જો શરત સાચી હોય તો "First Class with Distinction" સંદેશ દર્શાવી પ્રોગ્રામ પૂર્ણ કરવામાં આવે છે. જો પ્રથમ શરતનું પરિણામ False મળશે તો બીજી શરત ચકાસવામાં આવશે. જો બીજી શરત ખોટી હશે તો ત્રીજી ને એમ ક્રમાનુસાર પ્રવાહ આગળ વધશે. જો આપેલ ચારેય શરતોનું પરિણામ False મળશે તો તે કિસ્સામાં પૂર્વનિર્ધારિત else બ્લોકનો અમલ કરી "Fail" સંદેશ દર્શાવવામાં આવશે.

else-if લેડર વિધાનની વાક્યરચના (Syntax of else...if ladder statement)

```
if (test expression-1)  
    Statement block-1;  
  
else if (test expression-2)  
    Statement block-2;  
  
else if (test expression-3)  
    Statement block-3;  
  
.....  
  
.....  
  
else if (test expression-n)  
    Statement block-n;  
  
else  
    Default-statement-block;  
  
program-statement-x;
```

આ બંધારણને else-if લેડર તરીકે ઓળખવામાં આવે છે. લેડરમાં ટેસ્ટ એક્સપ્રેશનનું ઉપરથી નીચેના ક્રમમાં મૂલ્યાંકન કરવામાં આવે છે. જ્યારે કોઈ પણ ટેસ્ટ એક્સપ્રેશન True પરિણામે ત્યારે તેની સાથે સંકળાયેલા વિધાનોના બ્લોકનો અમલ કરવામાં આવે છે. ત્યાર પછી લેડરના બાકીના વિભાગને કુદાવીને નિયંત્રણ program-statement-x તરફ વાળવામાં આવે છે. જો તમામ ટેસ્ટ-એક્સપ્રેશનનું મૂલ્યાંકન Falseમાં પરિણમે તો નિયંત્રણને અંતિમ else વિભાગના Default-statement-block તરફ લઈ જવામાં આવશે. else-if લેડર વિધાનનો ફ્લોચાર્ટ આકૃતિ 13.11માં દર્શાવ્યો છે.



આકૃતિ 13.11 : else-if લેડર વિધાનનો ફ્લો-ચાર્ટ

સ્વિચ વિધાન (The switch statement)

પ્રોગ્રામના અમલીકરણ દરમિયાન વિધાનોના જુદા-જુદા બ્લોકને પસંદ કરવા માટે આપણે if-else વિધાનનો ઉપયોગ કર્યો. જો કે, જ્યારે અનેક if-else વિધાનોનો ઉપયોગ કરવામાં આવે ત્યારે પ્રોગ્રામ જટિલ બની જાય છે. પ્રોગ્રામ વાંચવામાં અને તેનો તર્ક ઉકેલવામાં મુશ્કેલી પડે છે. સી ભાષા પ્રોગ્રામને સરળ બનાવવા માટે switch નામના એક આંતરપ્રસ્થાપિત બહુમાર્ગીય (multiway) નિર્ણય માટેનું વિધાન પૂરું પાડે છે. જ્યારે આપેલ અનેક પસંદગીઓમાંથી કોઈ એક ક્રિયા પસંદ કરવાની હોય ત્યારે switch વિધાન ઘણું ઉપયોગી બને છે.

switch વિધાનની વાક્યરચના (Syntax of the switch statement)

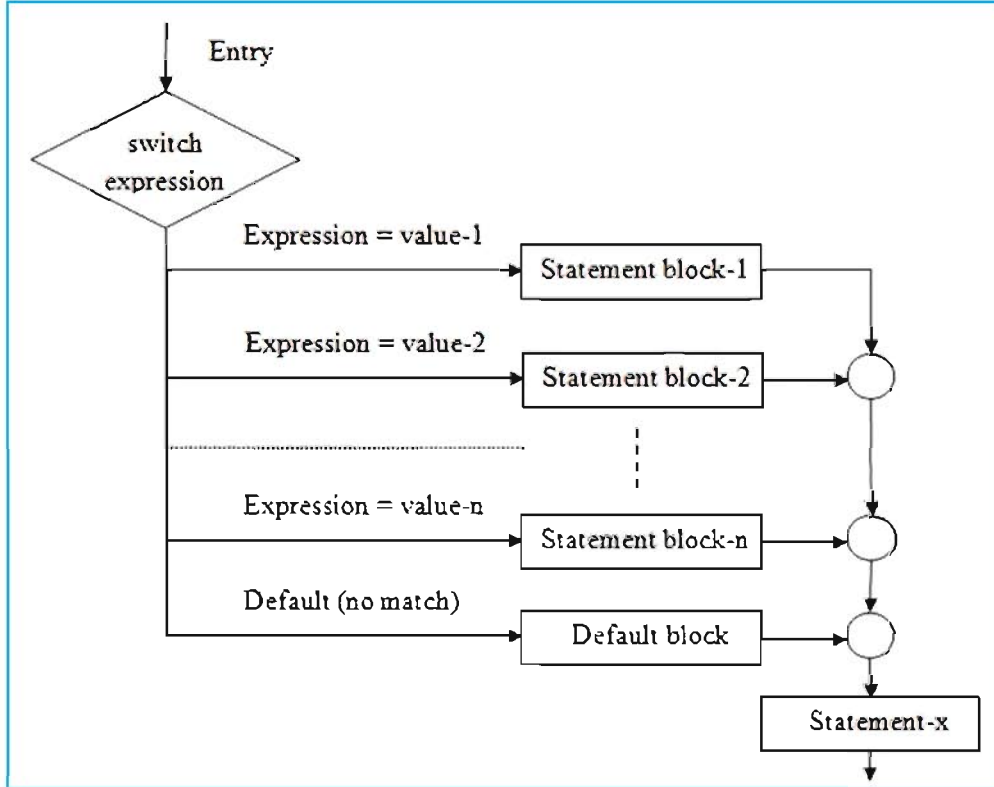
```

switch (expression)
{
    case value-1:
        statement block-1;
        break;
    case value-2:
        statement block-2;
        break;
    ....
    ....
    case value-n:
        statement block-n;
        break;
    default:
        default statement block;
        break;
}
statement-x;
  
```

switch વિધાનને સંબંધિત એવા નીચેના અગત્યના મુદ્દા નોંધી લો :

- switch વિધાન એક આર્ગ્યુમેન્ટ(પદાવલિ અથવા ચલના નામ)નો ઉપયોગ કરે છે. આ આર્ગ્યુમેન્ટને switch વિધાનની અંદર આવેલા અનેક કેસ વિકલ્પો સાથે સરખામણી માટે ચકાસવામાં આવે છે. આર્ગ્યુમેન્ટ એ પૂર્ણાંક કે અક્ષર ધરાવતો ચલ કે પદાવલિ હોઈ શકે.
- દરેક case વિકલ્પ અચળ કે અચળ પદાવલિ ધરાવે છે. આ અચળને કેસ લેબલ તરીકે ઓળખવામાં આવે છે તથા તેના અંતમાં વિસર્ગ (:) નિશાની મૂકવામાં આવે છે. દરેક કેસ લેબલ અનન્ય હોવા જરૂરી છે. કોઈ પણ બે કેસ લેબલની કિંમતો સમાન હોઈ શકે નહીં.
- Statement block-1, statement block-2 વગેરે વિધાનોની યાદી છે જેમાં શૂન્ય કે વધુ વિધાનોનો સમાવેશ થયેલો હોઈ શકે. આ વિધાનોના બ્લૉકને છગડિયા કૌંસમાં મૂકવાની જરૂર નથી.
- જ્યારે switch વિધાનનો અમલ કરવામાં આવે ત્યારે તે સૌપ્રથમ પદાવલિનું મૂલ્યાંકન કરે છે અને પછી ઉપરથી નીચેના ક્રમમાં કેસ અચળ સાથે તેની સરખામણી કરે છે. જ્યારે કિંમતની સરખામણી આપેલ પદાવલિ સાથે મળે ત્યારે તે કેસને સંબંધિત વિધાનોના વિભાગનો અમલ કરવામાં આવશે.
- switch વિધાનમાં દરેક કેસ વિભાગના વિધાનો પછી આપવામાં આવેલ break વિધાન જે-તે કેસનો અંત સૂચવે છે અને નિયંત્રણને switch વિધાનની બહાર લઈ જાય છે. ત્યાર પછી નિયંત્રણનો પ્રવાહ પ્રોગ્રામ પછીના વિધાન (statement-x) તરફ જાય છે.
- default એ મરજિયાત કેસ છે. અમલીકરણ દરમિયાન જ્યારે કોઈ પણ કેસની સરખામણી મેળવી શકાતી નથી, ત્યારે default વિધાનોના વિભાગનો અમલ કરવામાં આવે છે. default વિધાનનો ઉપયોગ એક જ વખત કરી શકાય છે તથા તેને switch વિધાનમાં કોઈ પણ સ્થાને લખી શકાય છે. પરંતુ સામાન્ય રીતે આપણે તેને switch વિધાનના અંત ભાગમાં ઉમેરતા હોઈએ છીએ.

switch વિધાનનું અમલીકરણ આકૃતિ 13.12માં દર્શાવેલ ફ્લો-ચાર્ટ દ્વારા સમજાવવામાં આવ્યું છે. આકૃતિમાં એવું માની લેવામાં આવ્યું છે કે દરેક કેસ વિભાગમાં અંતિમ વિભાગ તરીકે break વિધાન છે.



આકૃતિ 13.12 : switch વિધાનનો ફ્લો-ચાર્ટ

ઉદાહરણ 13.5 : હવે આપણે આપેલ એક અંક (0થી 9) પરથી તેને સુસંગત એવા શબ્દની રજૂઆત કરે તેવા સી પ્રોગ્રામને સમજાવે. આકૃતિ 13.13માં ઉદાહરણ 13.5નું કોડ-લિસ્ટિંગ આપેલું છે તથા આકૃતિ 13.14 તેનું પરિણામ દર્શાવે છે.

```
10-5.c - SciTE
File Edit Search View Tools Options Language Buffers Help
10-5.c
/* Example 5: Program to illustrate use of switch statement.*/
#include<stdio.h>
int main()
{
    int number;
    printf("Enter a single digit: ");           // message to the user
    scanf("%d", &number);                       // read a number from user
    switch (number)
    {
        case 0:
            printf("0 = Zero \n");             break;
        case 1:
            printf("1 = One \n");               break;
        case 2:
            printf("2 = Two \n");               break;
        case 3:
            printf("3 = Three \n");             break;
        case 4:
            printf("4 = Four \n");              break;
        case 5:
            printf("5 = Five \n");              break;
        case 6:
            printf("6 = Six \n");               break;
        case 7:
            printf("7 = Seven \n");             break;
        case 8:
            printf("8 = Eight \n");             break;
        case 9:
            printf("9 = Nine \n");              break;
        default:
            printf("Out of range number entered. \n");
    }
    return 0;
} /* End of Program */
```

આકૃતિ 13.13 : ઉદાહરણ 13.5નું કોડ-ચિત્રિંગ

```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 10-5.c
kpp@ubuntu:~$ ./a.out
Enter a single digit: 5
5 = Five
kpp@ubuntu:~$ ./a.out
Enter a single digit: 105
Out of range number entered.
kpp@ubuntu:~$
```

આકૃતિ 13.14 : ઉદાહરણ 13.5નું પરિણામ

સમજૂતી

main() વિધેયમાં પ્રથમ, બીજું અને ત્રીજું વિધાન અનુક્રમે ચલ ઘોષિત કરે છે, સંદેશ દર્શાવે છે તથા ઉપયોગકર્તા પાસેથી એક અંક મેળવે છે. ચોથા વિધાનમાં switch વિધાનની આર્ગ્યુમેન્ટ તરીકે number ચલમાં સંગૃહીત કિંમત અનુસાર સુસંગત કેસના વિભાગનો અમલ કરવામાં આવશે. જો કોઈ પણ caseમાં સુસંગતતા નહીં મળે તો default વિભાગ "Out of range number entered" સંદેશ દર્શાવશે અને પ્રોગ્રામ પૂરો થશે.

હવે, switch વિધાનમાં અક્ષર પ્રકારના અચળનો ઉપયોગ કેસ લેબલ તરીકે કેવી રીતે કરી શકાય તે ઉદાહરણ 13.6નો ઉપયોગ કરીને સમજાવે.

ઉદાહરણ 13.6 : ઉપયોગકર્તા પાસેથી બે અંકો મેળવી મેનૂની મદદથી ગાણિતિક પ્રક્રિયક(arithmetic operator)ની પસંદગી માંગો. આપેલ પસંદગી પ્રમાણે આપેલ અંકો પર ગાણિતિક પ્રક્રિયા કરી પરિણામ દર્શાવો. આકૃતિ 13.15માં ઉદાહરણ 13.6નું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 13.16 તેનું પરિણામ દર્શાવે છે.

```
10-6.c - SciTE
File Edit Search View Tools Options Language Buffers Help

/* Example 6: Program to illustrate use of switch statement using character case label.*/
#include<stdio.h>
#include<stdlib.h>
int main() {
    char choice;
    int number1, number2;
    system("clear");
    printf("Enter two numbers:");           // message to the user
    scanf("%d %d", &number1, &number2);    // read two numbers from user
    printf("\nMENU OF ARITHMETIC OPERATION \n");
    printf("A - Addition \n");
    printf("S - Subtraction \n");
    printf("M - Multiplication \n");
    printf("D - Division \n");
    printf("Enter your choice (A/S/M/D) : "); // message to the user
    scanf(" %c", &choice);                 // read choice from user
    switch (choice)
    {
        case 'A':
            printf("Addition of entered numbers: %d \n", number1 + number2);
            break;
        case 'S':
            printf("Subtraction of entered number: %d \n", number1 - number2);
            break;
        case 'M':
            printf("Multiplication of entered number: %d \n", number1 * number2);
            break;
        case 'D':
            printf("Division of entered number: %d \n", number1 / number2);
            break;
        default:
            printf("Invalid menu choice. \n");
            break;
    }
    return 0;
} /* End of Program */
```

આકૃતિ 13.15 : ઉદાહરણ 13.6નું કોડ-લિસ્ટિંગ

```
kpp@ubuntu: ~  
File Edit View Terminal Help  
Enter two numbers:20 30  
  
MENU OF ARITHMETIC OPERATION  
A - Addition  
S - Subtraction  
M - Multiplication  
D - Division  
Enter your choice (A/S/M/D) : A  
Addition of entered numbers: 50  
kpp@ubuntu:~$
```

આકૃતિ 13.16 : ઉદાહરણ 13.6નું પરિણામ

સમજૂતી

main() વિધેયનું ચોથું અને પાંચમું વિધાન અનુક્રમે સંદેશ દર્શાવશે અને ઉપયોગકર્તા પાસેથી બે કિંમતો મેળવશે. વિધાન છ થી દસ દ્વારા સ્ક્રીન પર મેનૂ વિકલ્પો દર્શાવવામાં આવશે. અગિયારમા અને બારમા વિધાનમાં અનુક્રમે સંદેશ દર્શાવશે અને ઉપયોગકર્તા પાસેથી મેનૂની પસંદગી મેળવવામાં આવશે. ઉપયોગકર્તાએ કરેલ પસંદગીને આધારે તેને સુસંગત કેસ વિભાગનો અમલ કરવામાં આવશે. જો કોઈ પણ કેસમાં સરખામણી નહીં મળે તો પૂર્વનિર્ધારિત (default) વિભાગ "Invalid menu choice" સંદેશ દર્શાવશે અને પ્રોગ્રામ પૂર્ણ થશે. આ પ્રોગ્રામમાં 'શૂન્ય વડે ભાગાકાર' જેવી ભૂલની સ્થિતિ તપાસવા માટે યોગ્ય જગ્યાએ વધારાનાં વિધાનો પણ ઉમેરી શકાય.

સંયોજિત વિધાનો (Compound statement)

સી ભાષામાં સંયોજન સંબંધિત ચકાસણીઓ (compound relational tests) અથવા એકથી વધુ પદાવલિઓની ચકાસણી (multiple test expression) પાર પાડવા માટેની જરૂરી સુવિધા પણ આપવામાં આવી છે. સંયોજન સંબંધિત ચકાસણીઓ એટલે એક કે વધુ સંબંધિત ચકાસણીઓને તાર્કિક AND અથવા તાર્કિક OR પ્રક્રિયક દ્વારા જોડવી. આ તાર્કિક પ્રક્રિયકો અનુક્રમે && અને || અક્ષર-યુગ્મ દ્વારા રજૂ કરવામાં આવે છે. સંયોજન સંબંધિત ચકાસણીઓ પ્રોગ્રામમાં આવેલ if...else વિધાનોની સંખ્યાને ઘટાડવામાં મદદરૂપ બને છે.

સારાંશ

આ પ્રકરણમાં આપણે સી ભાષાની વિશિષ્ટ પ્રકારની સુવિધાઓ વિશે અભ્યાસ કર્યો, કે જેના દ્વારા પ્રોગ્રામમાં સૂચનાઓના ક્રમનો પ્રવાહ બદલી શકાય છે. પ્રોગ્રામમાં આવેલ સૂચનાઓના ક્રમનો પ્રવાહ બદલવા માટે બે નિર્ણય માળખાં : if અને switchનો ઉપયોગ કેવી રીતે કરવો તે આપણે શીખ્યા.

સ્વાધ્યાય

1. if...else વિધાનની સરખામણીમાં if વિધાનની મર્યાદા જણાવો.
2. નેસ્ટેડ if (nested if) એટલે શું ?
3. switch વિધાનનો ઉપયોગ સલાહભર્યો ગણાય તેવું એક યોગ્ય ઉદાહરણ આપો.
4. switch બંધારણમાં breakનું મહત્વ જણાવો.
5. યોગ્ય ઉદાહરણ સાથે else...if લેડર વિધાન સમજાવો.

6. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો :

- (a) if વિધાનને switch વિધાનની અંદર લખી શકાય છે.
- (b) જ્યારે if વિધાનના ટેસ્ટ-એક્સપ્રેશનનું પરિણામ False આવે છે ત્યારે ifની અંદર આવેલાં વિધાનોને અવગણવામાં આવે છે.
- (c) switch વિધાનમાં default વિધાન ફરજિયાત છે.
- (d) switch વિધાનમાં default વિધાન હંમેશા અંતિમ વિધાન હોવું જોઈએ.
- (e) break વિધાન પ્રોગ્રામના અમલીકરણને અટકાવે છે.
- (f) if વિધાનની અંદર switch વિધાનને ઉમેરી શકાય છે.

7. આપેલ વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) નીચેના if વિધાનના અમલ બાદ flag ચલની કિંમત કઈ હશે ?

```
int flag=0;
```

```
if(5 < 8) {flag=1;}
```

- (a) 0 (b) 1 (c) 5 (d) 8

(2) નીચેના switch વિધાનના અમલ પછી ચલ 's'ની કિંમત કઈ હશે ?

```
x = 3;
```

```
switch (x) {
```

```
    case 1 : s = 'A'; break;
```

```
    case 2 : s = 'B'; break;
```

```
    case 3 : s = 'C'; break;
```

```
    default : s = 'D'; break;
```

```
}
```

- (a) A (b) B (c) C (d) D

(3) નીચેનાં વિધાનોના અમલ બાદ ચલ sumની કિંમત કઈ હશે ?

```
int number1=5, number2=10, sum=0;
```

```
if (number1 > number2)
```

```
    sum = sum + number1;
```

```
else
```

```
    sum = sum + number2;
```

- (a) 0 (b) 5 (c) 10 (d) 12

(4) નીચે આપેલા પ્રોગ્રામ-ખંડનો અમલ કરવામાં આવે તો શું પરિણામ મળશે ?

```
int number1=10, number2=20;
if ((number1+number2) > 35 || (number1>number2))
    printf("%d", number1);
else
    printf("%d", number2);
```

- (a) 10 (b) 20 (c) 35 (d) ભૂલનો સંદેશ

(5) નીચે આપેલા પ્રોગ્રામ-ખંડનો અમલ કરવામાં આવે તો શું પરિણામ મળશે ?

```
char chr='A';
switch (chr)
{
    case 'A' : printf("A"); break;
    case 'B' : printf("B"); break;
    case 'C' : printf("C"); break;
}
```

- (a) A (b) B (c) C (d) ભૂલનો સંદેશ

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો માટે સી પ્રોગ્રામ લખો :

1. આપેલ સંખ્યા ધન છે કે ઋણ તે તપાસો.
2. વ્યક્તિની ઉંમર મેળવી તે મતાધિકારને પાત્ર છે કે નહીં તે દર્શાવો.
3. ઉપયોગકર્તા પાસેથી ત્રણ સંખ્યાઓ મેળવી નેસ્ટેડ ifની મદદથી તેમાંથી નાનામાં નાની સંખ્યા શોધો.
4. આપેલ સંખ્યા એકી છે કે બેકી તે તપાસો.
5. switch વિધાનની મદદથી આપેલ અક્ષર સ્વર છે કે નહીં તે ચકાસો.





લૂપ નિયંત્રણ માળખાં

સી પ્રોગ્રામમાં ક્યારેક એવું બની શકે કે જેમાં એકસરખાં વિધાનોના ખંડને એકથી વધુ વખત અમલમાં મૂકવાનો હોય. એક વિધાન કે વિધાનોના સમૂહને એકથી વધુ વાર અમલમાં મૂકવા માટે પ્રોગ્રામરોને તમામ પ્રોગ્રામિંગ ભાષાઓ દ્વારા લૂપ નિયંત્રણ માળખાં (જે **લૂપિંગ / looping** તરીકે પણ ઓળખાય છે તે) પૂરાં પાડવામાં આવે છે. આ પ્રકરણમાં આપણે સી પ્રોગ્રામિંગ ભાષા દ્વારા પૂરા પાડવામાં આવતા લૂપ (પુનરાવર્તિત) નિયંત્રણ માળખાં વિશે ચર્ચા કરીશું. અહીં નિયંત્રણ માળખાં(control structure)નો અર્થ એ થાય છે કે અમલીકરણનો પ્રવાહ ક્રમબદ્ધ હોવો જરૂરી નથી તથા પ્રોગ્રામમાં આપવામાં આવેલી શરત અનુસાર નિયંત્રણ કોઈપણ વિધાન તરફ મોકલી શકાય છે.

લૂપિંગમાં કોઈ નિર્ગમ શરત (exit condition) ન સંતોષાય ત્યાં સુધી વિધાનોની શ્રેણીનો અમલ કરવામાં આવે છે. લૂપિંગ માળખાં બે વિભાગોમાં રચવામાં આવે છે : લૂપનો મુખ્ય ભાગ (body of loop) અને નિયંત્રણ વિધાનો (control statement). નિયંત્રણ વિધાનના સ્થાનને આધારે લૂપને **પ્રવેશ નિયંત્રણ (entry controlled)** લૂપ અને **નિર્ગમ નિયંત્રણ (exit controlled)** લૂપ એમ બે પ્રકારોમાં વર્ગીકૃત કરી શકાય. નિવેશ નિયંત્રણ લૂપમાં લૂપના મુખ્ય ભાગમાં આવેલાં વિધાનોના અમલ પહેલાં નિર્ગમ / નિયંત્રણ શરત ચકાસવામાં આવે છે. જ્યારે, નિર્ગમ નિયંત્રણ લૂપમાં લૂપના મુખ્ય ભાગમાં આવેલાં વિધાનોના અમલ બાદ નિર્ગમ/નિયંત્રણ શરતને ચકાસવામાં આવે છે. આનો અર્થ એ થાય કે નિર્ગમ નિયંત્રણ લૂપમાંથી બહાર નીકળતાં પહેલાં લૂપનાં વિધાનોનો અમલ ઓછામાં ઓછી એક વાર તો કરવામાં આવશે જ. આ માટેનું વધુ વિસ્તૃત વિવરણ જુદા જુદા પ્રકારના લૂપની સમજૂતી વખતે આપવામાં આવ્યું છે.

હવે, સી ભાષામાં ઉપલબ્ધ નીચેનાં ત્રણ લૂપ નિયંત્રણ માળખાં સંબંધિત વાક્યરચના અને સામાન્ય નિયમો વિશે ચર્ચા કરીએ :

- for
- while
- do...while

for લૂપ (The for loop)

વિધાનોના સમૂહનો નિશ્ચિત સમય સુધી અમલ કરવા માટે સામાન્ય રીતે *for* લૂપનો ઉપયોગ કરવામાં આવે છે. *for* લૂપને વધુ ક્રિયાશીલ (dynamic) બનાવવા માટે તેમાં નિર્ગમ શરતનો ઉપયોગ કરી શકાય છે. ઉદાહરણ 14.1 દ્વારા સરળ *for* લૂપ વિધાનનો ઉપયોગ સમજવાનો પ્રયત્ન કરીએ. આ ઉદાહરણ સરળ *for* લૂપ વિધાનનો ઉપયોગ કરી પાંચ " * " દર્શાવે છે. આકૃતિ 14.1માં ઉદાહરણ 14.1નું કોડ લિસ્ટિંગ અને પરિણામ આપવામાં આવ્યું છે.

```
11-1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

11-1.c

/* Example 1: Program to illustrate simple for loop statement.*/
#include<stdio.h>

int main()
{
    int count;                                // declaration of variables
    for ( count = 0; count < 5 ; count++) // for loop body repeats 5 times
        printf(" * ");
    return 0;
}
/*End of Program */

gcc 11-1.c
>gcc 11-1.c
>Exit code: 0
./a.out
>./a.out
* * * * * >Exit code: 0
```

આકૃતિ 14.1 : ઉદાહરણ 14.1નું કોડ-લિસ્ટિંગ અને પરિણામ

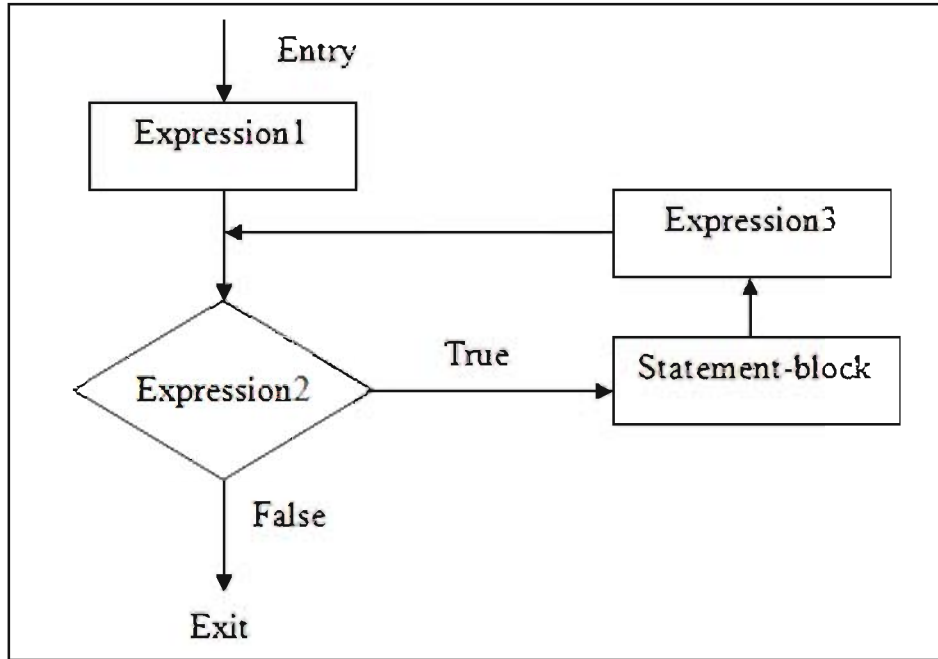
સમજૂતી

પ્રોગ્રામની શરૂઆતમાં count નામનો એક પૂર્ણાંક ચલ ઘોષિત કરવામાં આવ્યો છે. for લૂપ માટે count ચલનો ઉપયોગ ગણતરી માટેના ચલ (counter variable) તરીકે કરવામાં આવ્યો છે. શરૂઆતમાં count ચલની કિંમત શૂન્ય આપવામાં આવી છે. ત્યારબાદ બીજી પદાવલિ (count < 5) ને ચકાસવામાં આવે છે, જો તેનું મૂલ્યાંકન સાચું (true) થાય તો for લૂપનાં વિધાનોનો અમલ કરવામાં આવે છે. આ ઉદાહરણમાં અહીં printf વિધાનની મદદથી સ્ક્રીન પર એક " * " દર્શાવવામાં આવશે. ત્યારપછી for લૂપની ત્રીજી પદાવલિ (count++) નો અમલ કરી count ની કિંમતમાં 1 નો વધારો કરવામાં આવશે. ફરી બીજી પદાવલિ (count < 5) નું મૂલ્યાંકન કરવામાં આવશે અને તેના પરિણામને આધારે for લૂપનાં વિધાનોનો અમલ કરવામાં આવશે. આ પ્રોગ્રામમાં printf વિધાનનો અમલ પાંચ વખત કરવામાં આવશે અને સ્ક્રીન પર પાંચ વાર ફૂદડીની નિશાની " * * * * * " દર્શાવવામાં આવશે.

for લૂપની વાક્યરચના (Syntax of for loop)

```
for (expression1; expression2; expression3)
{
    statement-block;
}
```

for લૂપના મથાળામાં અર્ધવિરામથી જુદી પાડેલી ત્રણ પદાવલિઓ કૌંસમાં મૂકવામાં આવે છે. આ તમામ પદાવલિઓ મરજિયાત છે. સ્ટેટમેન્ટ બ્લોકના નામે ઓળખાતા for લૂપના મુખ્ય ભાગમાં એક વિધાન કે એકથી વધુ સંમિશ્રિત (compound) વિધાનો હોઈ શકે છે. નીચે આપેલ આકૃતિ 14.2 માં for લૂપનો ફ્લો-ચાર્ટ દર્શાવ્યો છે :



આકૃતિ 14.2 : for લૂપનો ફ્લો ચાર્ટ

આકૃતિ 14.2 માં દર્શાવ્યા મુજબ for લૂપના અમલીકરણનાં સોપાન નીચે પ્રમાણે છે :

સોપાન 1 : પદાવલિ-1(expression1) નું મૂલ્યાંકન કરો. for લૂપ શરૂ કરવામાં આવે ત્યારે પદાવલિ-1 નો અમલ એક જ વાર કરવામાં આવે છે.

સોપાન 2 : પદાવલિ-2 નું મૂલ્યાંકન કરો. જો પરિણામ ખોટું (false) મળે તો લૂપને અટકાવો.

સોપાન 3 : જો પદાવલિ-2 નું પરિણામ સાચું (true) મળે તો લૂપનાં વિધાનોનો અમલ કરો.

સોપાન 4 : પદાવલિ-3 નું મૂલ્યાંકન કરો.

સોપાન 5 : સોપાન 2 પર જાઓ.

for લૂપને સંબંધિત નીચેના મહત્વના મુદ્દાઓની નોંધ કરો :

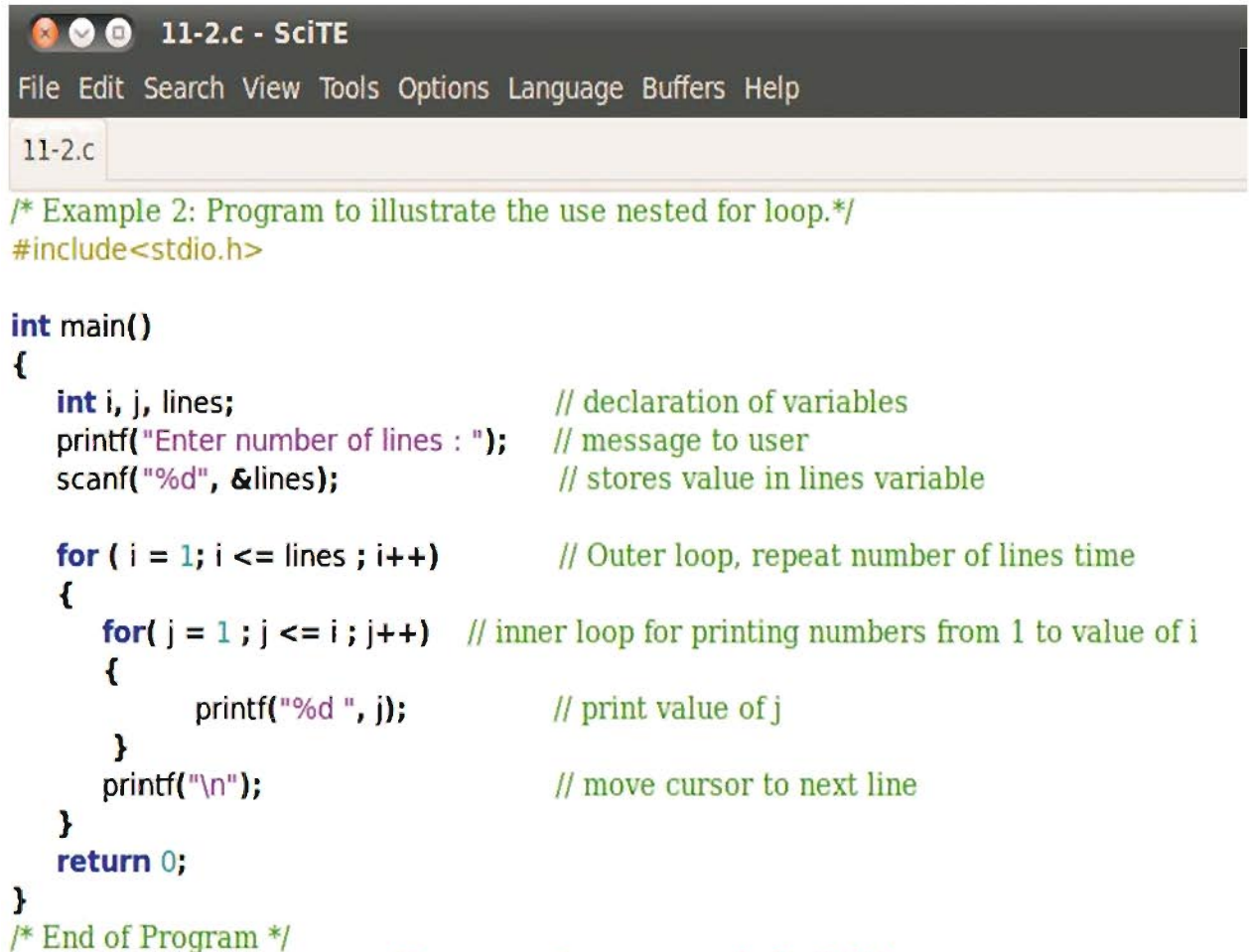
- કાઉન્ટર ચલની શરૂઆતની કિંમત આપવા માટે પદાવલિ-1નો ઉપયોગ કરવામાં આવે છે. કાઉન્ટર ચલનો ઉપયોગ લૂપના અમલીકરણની કુલ સંખ્યાનું નિયંત્રણ કરે છે. આ ચલને નિયંત્રણ ચલ (counter variable) તરીકે ઓળખવામાં આવે છે.
- પદાવલિ-2 ચકાસણીની શરત (ટેસ્ટ કન્ડિશન) તરીકે કાર્ય કરે છે. લૂપને અટકાવવા માટેના માપદંડને તપાસવા માટે અહીં નિયંત્રણ ચલનો ઉપયોગ કરવામાં આવે છે.
- નિયંત્રણ ચલની કિંમત વધારવા કે ઘટાડવા માટે પદાવલિ-3નો ઉપયોગ કરવામાં આવે છે.

નેસ્ટેડ for લૂપ (Nested for loop)

એક for લૂપની અંદર અન્ય for લૂપના ઉપયોગને નેસ્ટેડ for (nested for) કહે છે. આ અભિગમ સમજવા માટે ઉદાહરણ 14.2માં આવેલ પ્રોગ્રામનો ઉપયોગ કરીએ. આ ઉદાહરણ આપેલ લીટીની સંખ્યા પ્રમાણે નીચે દર્શાવેલ ભાત (pattern) દર્શાવશે. ઉદાહરણ તરીકે, જો લીટીની સંખ્યા 4 આપવામાં આવે તો પ્રોગ્રામ નીચે દર્શાવેલ પરિણામ આપશે :

```
1
1 2
1 2 3
1 2 3 4
```

આકૃતિ 14.3માં ઉદાહરણ 14.2નું કોડ લિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 14.4 તેનું પરિણામ દર્શાવે છે.



```
11-2.c - SciTE
File Edit Search View Tools Options Language Buffers Help

11-2.c

/* Example 2: Program to illustrate the use nested for loop.*/
#include<stdio.h>

int main()
{
    int i, j, lines;           // declaration of variables
    printf("Enter number of lines : "); // message to user
    scanf("%d", &lines);      // stores value in lines variable

    for ( i = 1; i <= lines ; i++) // Outer loop, repeat number of lines time
    {
        for( j = 1 ; j <= i ; j++) // inner loop for printing numbers from 1 to value of i
        {
            printf("%d ", j);      // print value of j
        }
        printf("\n");              // move cursor to next line
    }
    return 0;
}

/* End of Program */
```

આકૃતિ 14.3 : ઉદાહરણ 14.2નું કોડ-લિસ્ટિંગ

```
kpp@ubuntu: ~  
File Edit View Terminal Help  
kpp@ubuntu:~$ gcc 11-2.c  
kpp@ubuntu:~$ ./a.out  
Enter number of lines : 4  
1  
1 2  
1 2 3  
1 2 3 4  
kpp@ubuntu:~$
```

આકૃતિ 14.4 : ઉદાહરણ 14.2નું પરિણામ

સમજૂતી

પ્રોગ્રામમાં બહારનું લૂપ નિયંત્રણ ચલ i ધરાવે છે, જેની કિંમત એક પછી એક વધારતા જઈ 'lines' ચલની કિંમત જેટલી કરવામાં આવે છે. બહારના લૂપના i ની દરેક કિંમત માટે અંદરના લૂપમાં નિયંત્રણ ચલ j નો i વખત અમલ કરવામાં આવશે. અને j ની કિંમત 1થી શરૂ કરી એક પછી એક i સુધી વધારો કરી દર્શાવવામાં આવશે. i જેટલા અંકોને i જેટલી લીટીમાં દર્શાવ્યા પછી બહારના લૂપમાં આવેલું `printf("\n")` વિધાન કર્સરને નીચેની લીટીમાં લાવશે. તેથી ત્યાર પછીના અંકો નવી લીટીમાં દર્શાવશે.

for લૂપમાં કોમા પ્રક્રિયકનો ઉપયોગ (Use of comma operator in for loop)

for લૂપમાં કોમા પ્રક્રિયકનો ઉપયોગ કરી આપણે એકથી વધુ પ્રાયલનો આરંભ કરી શકીએ છીએ. આ જ રીતે for લૂપમાં એકથી વધુ ચલોની કિંમત વધારવા કે ઘટાડવા માટે પણ કોમા પ્રક્રિયકનો ઉપયોગ કરી શકાય છે. ઉદાહરણ 14.3ની મદદથી કોમા પ્રક્રિયકના ઉપયોગને સમજવાનો પ્રયત્ન કરીએ. આકૃતિ 14.5માં ઉદાહરણ 14.3નું કોડ ટિસ્ટિંગ આપવામાં આવ્યું છે તથા આકૃતિ 14.6 તેનું પરિણામ દર્શાવે છે.

```
11-3.c - SciTE  
File Edit Search View Tools Options Language Buffers Help  
11-3.c  
/* Example 3: Program to illustrate the use comma operator in a for loop  
to print the addition table.*/  
  
#include<stdio.h>  
int main()  
{  
    int i, j, number; // declaration of variables  
    printf("Enter number of lines in a table: "); // message to user  
    scanf("%d", &number); // stores value in number variable  
    for ( i=0, j=10; i < number; i++, j- )  
    {  
        printf("%d + %d = %d\n", i, j, i + j);  
    }  
    return 0;  
}  
/* End of Program */
```

આકૃતિ 14.5 : ઉદાહરણ 14.3નું કોડ-ટિસ્ટિંગ


```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 11-3.c
kpp@ubuntu:~$ ./a.out
Enter number of lines in a table: 5
0 + 10 = 10
1 + 9 = 10
2 + 8 = 10
3 + 7 = 10
4 + 6 = 10
kpp@ubuntu:~$ 

```

આકૃતિ 14.6 : ઉદાહરણ 14.3નું પરિણામ

ઉદાહરણ 14.3માં $i=0$ અને $j=10$ કિંમતો સાથે બે ચલનો ઉપયોગ કરવામાં આવ્યો છે. વળી, એક જ *for* લૂપમાં i ની કિંમત વધારવામાં અને j ની કિંમત ઘટાડવામાં આવી છે. આપવામાં આવેલ ઇનપુટને આધારે ઉદાહરણમાં દર્શાવ્યા મુજબ પરિણામ દર્શાવવામાં આવે છે.

while લૂપ (The while loop)

જ્યારે ફેરાની સંખ્યા પૂર્વનિશ્ચિત ન હોય અને લૂપને અટકાવવા માટેની શરત લૂપમાં પ્રવેશ કરતાં પહેલાં ચકાસવાની હોય ત્યારે *while* લૂપનો ઉપયોગ વધુ અનુકૂળ છે. ઉદાહરણ તરીકે,

- ઉપયોગકર્તા શૂન્ય ન ઉમેરે ત્યાં સુધી ઉમેરવામાં આવેલી તમામ સંખ્યાઓનો સરવાળો કરવો.
- મેનૂ વિકલ્પો દર્શાવવા અને ઉપયોગકર્તા બહાર નીકળવાનો (exit) વિકલ્પ પસંદ ન કરે ત્યાં સુધી યોગ્ય ક્રિયાઓનો અમલ કરવો.

while લૂપની વાક્યરચના (Syntax of while loop)

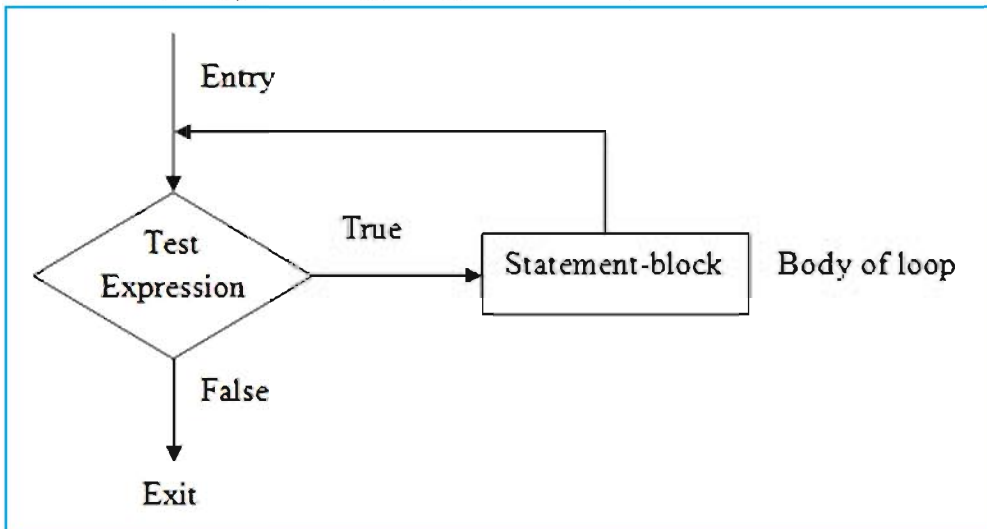
while (test expression)

```

{
    statement-block; /* while લૂપનાં વિધાનો */
}

```

while લૂપના અમલીકરણનો ફ્લો-ચાર્ટ આકૃતિ 14.7માં દર્શાવ્યો છે.

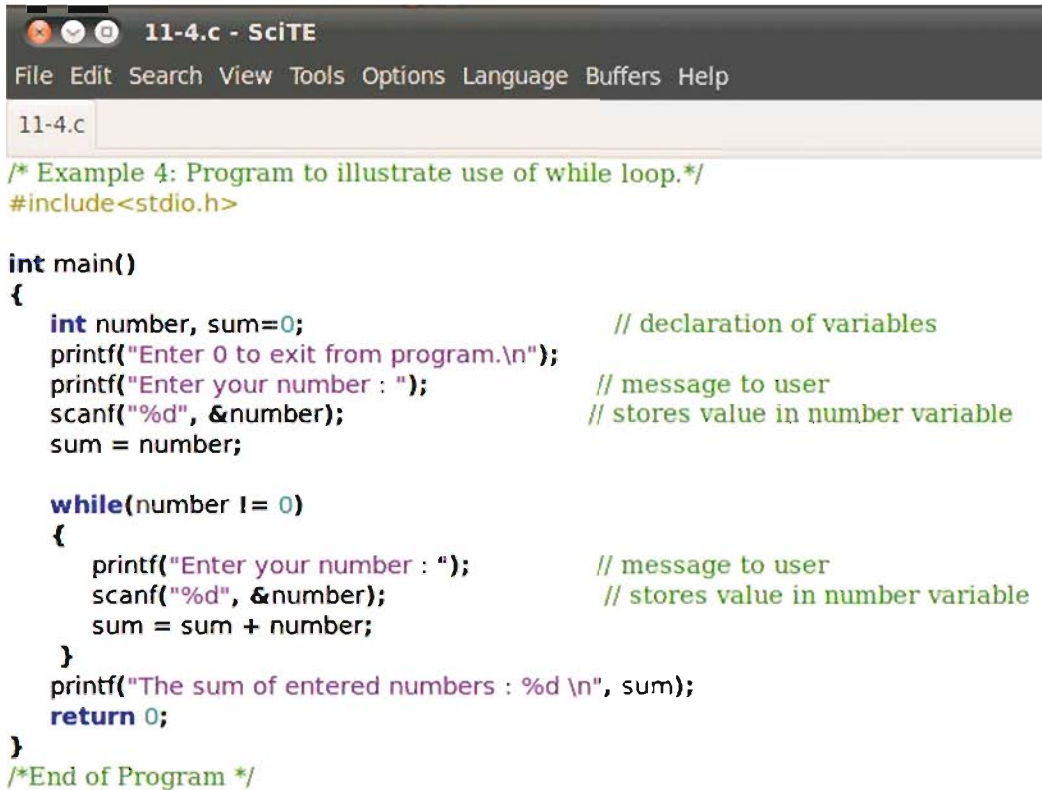


આકૃતિ 14.7 : *while* લૂપનો ફ્લો-ચાર્ટ

આકૃતિ 14.7માં દર્શાવ્યા મુજબ સૌ પ્રથમ શરત(ટેસ્ટ એક્સપ્રેશન)નું મૂલ્યાંકન કરવામાં આવશે. જો શરત true પરત કરશે તો લૂપનાં વિધાનોનો વિભાગ અમલી બનશે. પ્રોગ્રામ ફરી શરતનું મૂલ્યાંકન કરશે. શરતનું પરિણામ false ન મળે ત્યાં સુધી આ પ્રક્રિયાનું પુનરાવર્તન કરવામાં આવશે. જ્યારે શરતનું મૂલ્યાંકન false મળશે ત્યારે લૂપ અટકાવવામાં આવશે અને નિયંત્રણને પ્રોગ્રામમાં લૂપ પછી આપેલા વિધાન તરફ મોકલવામાં આવશે. લૂપનાં વિધાનો એક અથવા સંમિશ્રિત (compound) હોઈ શકે છે.

અહીં એ નોંધ લેવી જરૂરી છે કે, લૂપના પ્રવેશ ઉપર શરતની ચકાસણી થતી હોવાથી તેને પ્રવેશ નિયંત્રણ (entry controlled) લૂપ તરીકે ઓળખવામાં આવે છે. જો શરતનું પરિણામ પ્રથમ વખત જ false મળશે તો લૂપનાં વિધાનો એકપણ વાર અમલમાં મુકાશે નહીં.

હવે, ઉદાહરણ 14.4ની મદદથી while લૂપનો ઉપયોગ સમજવાનો પ્રયત્ન કરીએ. ઉપયોગકર્તા શૂન્ય ન ઉમેરે ત્યાં સુધી આપવામાં આવેલ તમામ સંખ્યાઓનો સરવાળો શોધવા માટેનો પ્રોગ્રામ ઉદાહરણ 14.4માં આપવામાં આવ્યો છે. આકૃતિ 14.8માં ઉદાહરણ 14.4નું કોડ-લિસ્ટિંગ આપવામાં આવ્યું છે જ્યારે આકૃતિ 14.9 તેનું પરિણામ દર્શાવે છે.



```
11-4.c - SciTE
File Edit Search View Tools Options Language Buffers Help

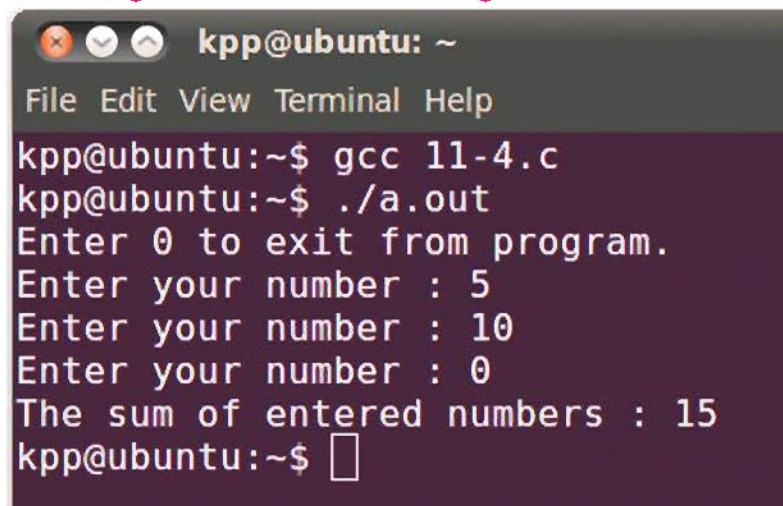
11-4.c

/* Example 4: Program to illustrate use of while loop.*/
#include<stdio.h>

int main()
{
    int number, sum=0; // declaration of variables
    printf("Enter 0 to exit from program.\n");
    printf("Enter your number : "); // message to user
    scanf("%d", &number); // stores value in number variable
    sum = number;

    while(number != 0)
    {
        printf("Enter your number : "); // message to user
        scanf("%d", &number); // stores value in number variable
        sum = sum + number;
    }
    printf("The sum of entered numbers : %d \n", sum);
    return 0;
}
/*End of Program */
```

આકૃતિ 14.8 : ઉદાહરણ 14.4નું કોડ-લિસ્ટિંગ



```
kpp@ubuntu: ~
File Edit View Terminal Help

kpp@ubuntu:~$ gcc 11-4.c
kpp@ubuntu:~$ ./a.out
Enter 0 to exit from program.
Enter your number : 5
Enter your number : 10
Enter your number : 0
The sum of entered numbers : 15
kpp@ubuntu:~$
```

આકૃતિ 14.9 : ઉદાહરણ 14.4નું પરિણામ

સમજૂતી

પ્રોગ્રામમાં જરૂરી એવા બે ચલની ઘોષણા `main()` વિધેય પછીના પ્રથમ વિધાન દ્વારા કરવામાં આવે છે. બીજા વિધાન દ્વારા પ્રોગ્રામમાંથી બહાર નીકળવાની રીતનો નિર્દેશ કરવામાં આવે છે. ત્રીજા અને ચોથા વિધાન દ્વારા અનુક્રમે સંદેશ દર્શાવી ઉપયોગકર્તા પાસેથી સંખ્યા મેળવવામાં આવે છે. પાંચમું વિધાન `number` ચલની કિંમત `sum` ચલને આપે છે. છઠ્ઠા વિધાનમાં શરત દ્વારા ચકાસવામાં આવે છે કે `number` ચલની કિંમત શૂન્ય છે કે નહીં. શરત સાચી હોય તો સાતમા અને આઠમા વિધાન દ્વારા અનુક્રમે સંદેશ દર્શાવી ઉપયોગકર્તા પાસેથી એક સંખ્યા મેળવવામાં આવે છે. દસમું વિધાન આપેલ સંખ્યાને `sum` ચલમાં ઉમેરે છે. ત્યાર પછી ફરી `while` લૂપની શરતનું મૂલ્યાંકન કરવામાં આવે છે. શરતનું પરિણામ નકારાત્મક (`false`) મળે ત્યારે ઉપયોગકર્તા દ્વારા ઉમેરવામાં આવેલી તમામ સંખ્યાઓનો સરવાળો દસમા વિધાનના અમલ દ્વારા દર્શાવવામાં આવે છે.

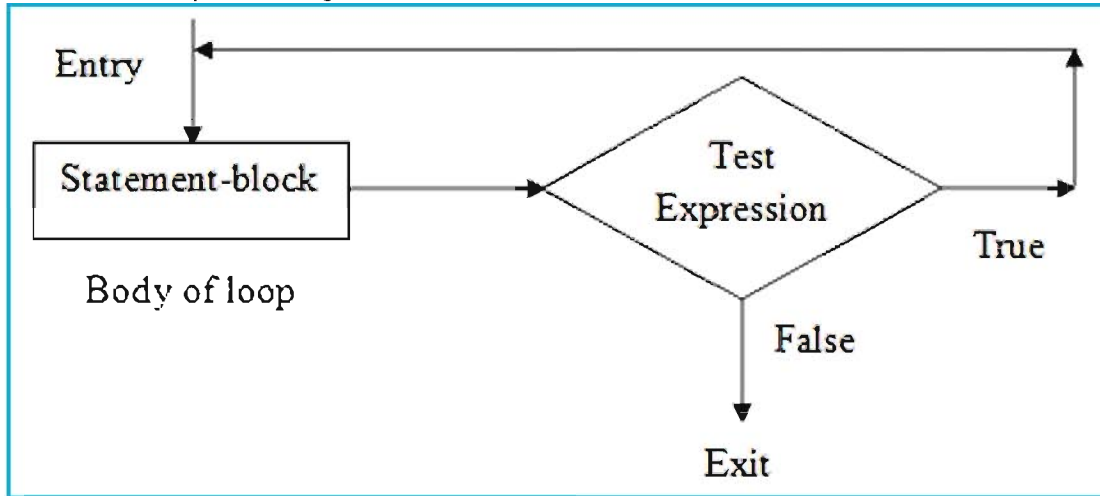
do...while લૂપ (The do...while loop)

આપણે જોયું કે, `while` લૂપમાં લૂપનાં વિધાનોનો અમલ કરતાં પહેલાં શરત ચકાસવામાં આવે છે. કેટલીકવાર આપણે શરતને ચકાસતાં પહેલાં લૂપનાં વિધાનોનો અમલ કરવા ઇચ્છતા હોઈએ છીએ. લૂપનાં વિધાનોનો અમલ કર્યા બાદ શરતને ચકાસવાની હોય ત્યારે `do...while` લૂપનો ઉપયોગ કરવો જોઈએ. લૂપના અંતમાં શરતની ચકાસણી કરવામાં આવતી હોવાથી `do...while` લૂપને નિર્ગમ-નિયંત્રણ (*exit controlled*) પ્રકારનું લૂપ કહે છે. અહીં એ નોંધ લેવી જરૂરી છે કે `do...while` લૂપમાં લૂપનાં વિધાનોનો અમલ એકવાર તો કરવામાં આવશે જ.

do...while લૂપની વાક્યરચના (Syntax of do...while loop)

```
do
{
    statement-block;    /* લૂપનાં વિધાનો */
}
while (tests expression);
```

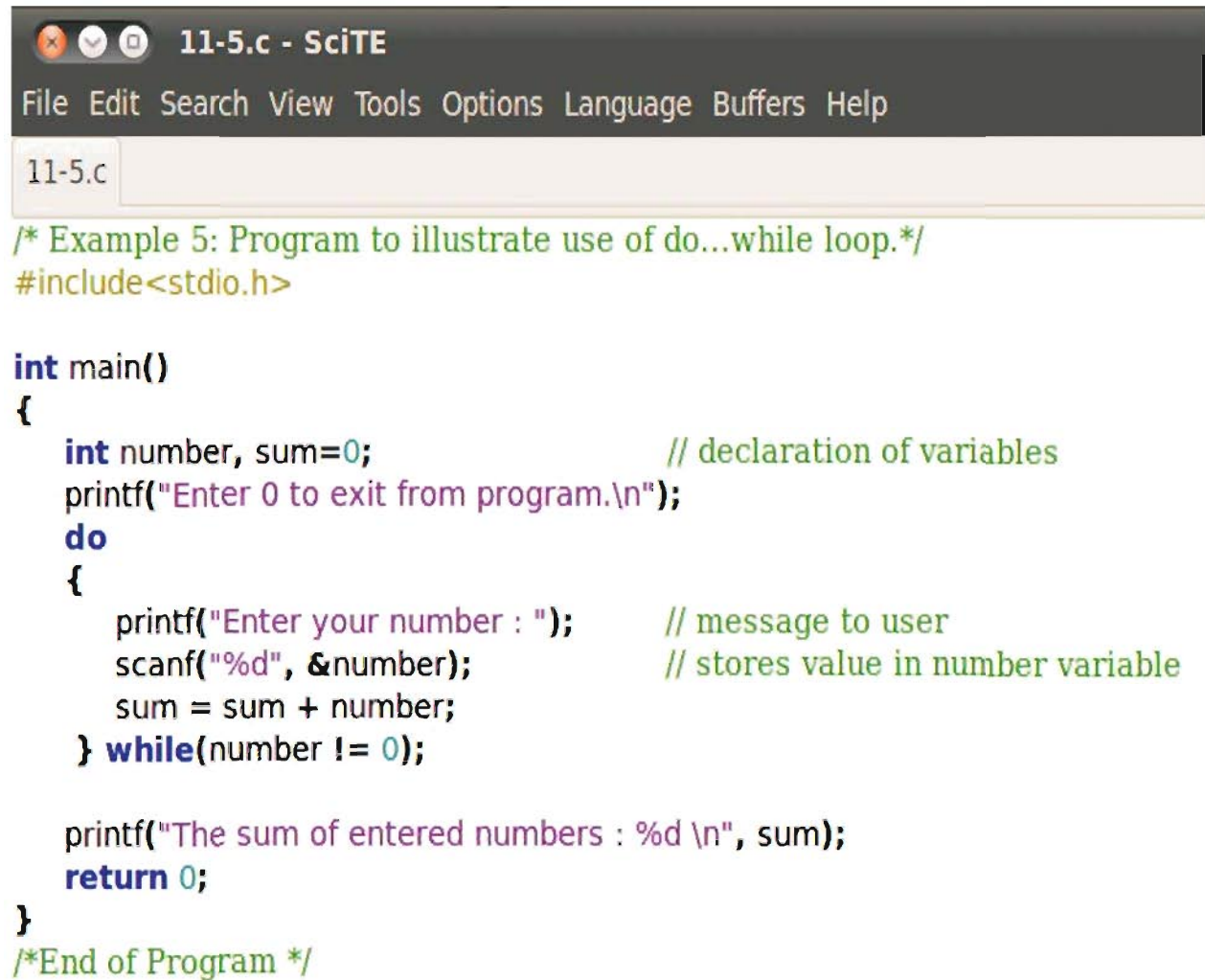
`do...while` લૂપનો ફ્લો-ચાર્ટ આકૃતિ 14.10માં દર્શાવ્યો છે.



આકૃતિ 14.10 : do...while લૂપનો ફ્લો-ચાર્ટ

આકૃતિ 14.10માં દર્શાવ્યા મુજબ લૂપમાં આવેલ સ્ટેટમેન્ટ બ્લોકનો પ્રથમ અમલ કરવામાં આવે છે. ત્યાર પછી શરતનું મૂલ્યાંકન કરવામાં આવે છે. જો શરતનું પરિણામ `true` મળે તો સ્ટેટમેન્ટ બ્લોક ધરાવતા લૂપનાં વિધાનોનો ફરી અમલ કરવામાં આવે છે. સ્ટેટમેન્ટ-બ્લોકના અમલ પછી ફરી એકવાર શરતનું મૂલ્યાંકન કરવામાં આવે છે. શરતનું પરિણામ `false` ન મળે ત્યાં સુધી આ ક્રિયાનું પુનરાવર્તન કરવામાં આવે છે. જ્યારે શરતનું પરિણામ નકારાત્મક આવે ત્યારે લૂપને અટકાવવામાં આવે છે અને નિયંત્રણને લૂપ પછીના વિધાન પર લઈ જવામાં આવે છે. લૂપમાં એક અથવા સંમિશ્રિત (*compound*) વિધાનો હોઈ શકે.

આકૃતિ 14.5નો ઉપયોગ કરી `do...while` લૂપની કાર્યપદ્ધતિ સમજવાનો પ્રયત્ન કરીએ. ઉપયોગકર્તા શૂન્ય ન ઉમેરે ત્યાં સુધી ઉમેરવામાં આવેલ તમામ સંખ્યાઓનો સરવાળો કરવા માટે `do...while` લૂપનો ઉપયોગ ઉદાહરણ 14.5માં કરવામાં આવ્યો છે. ઉદાહરણ 14.5નું કોડ લિસ્ટિંગ આકૃતિ 14.11માં આપવામાં આવ્યું છે તથા આકૃતિ 14.12 તેનું પરિણામ દર્શાવે છે.



```

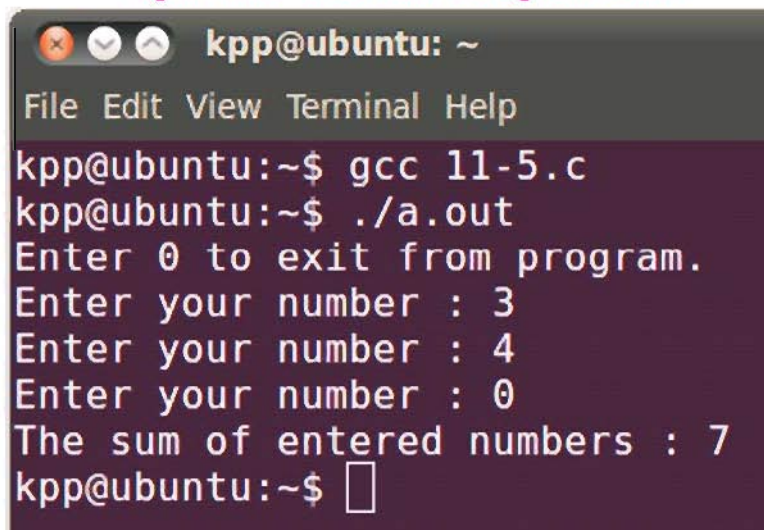
11-5.c
/* Example 5: Program to illustrate use of do...while loop.*/
#include<stdio.h>

int main()
{
    int number, sum=0;                // declaration of variables
    printf("Enter 0 to exit from program.\n");
    do
    {
        printf("Enter your number : ");    // message to user
        scanf("%d", &number);              // stores value in number variable
        sum = sum + number;
    } while(number != 0);

    printf("The sum of entered numbers : %d \n", sum);
    return 0;
}
/*End of Program */

```

આકૃતિ 14.11 : ઉદાહરણ 14.5નું કોડ-લિસ્ટિંગ



```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 11-5.c
kpp@ubuntu:~$ ./a.out
Enter 0 to exit from program.
Enter your number : 3
Enter your number : 4
Enter your number : 0
The sum of entered numbers : 7
kpp@ubuntu:~$ 

```

આકૃતિ 14.12 : ઉદાહરણ 14.5નું પરિણામ

સમજૂતી

main() વિધેય પછીનું પ્રથમ વિધાન પ્રોગ્રામમાં જરૂરી એવા બે ચલ ઘોષિત કરી તેમાંના એક ચલમાં કિંમત ઉમેરે છે. બીજું વિધાન પ્રોગ્રામમાંથી બહાર નીકળવાની રીતનો નિર્દેશ કરે છે. ત્રીજા વિધાનથી *do...while* લૂપની શરૂઆત થાય છે. ચોથું અને પાંચમું વિધાન અનુક્રમે સંદેશ દર્શાવે છે અને ઉપયોગકર્તા પાસેથી કિંમત મેળવે છે. છઠ્ઠું વિધાન ઉપયોગકર્તાએ આપેલ સંખ્યા *sum* ચલમાં ઉમેરે છે. સાતમું વિધાન શરતનું મૂલ્યાંકન કરે છે. જ્યાં સુધી શરત સાચી હોય ત્યાં સુધી ચોથાથી છઠ્ઠા વિધાનોનું પુનરાવર્તન કરવામાં આવે છે. જ્યારે શરતનું પરિણામ *false* મળે ત્યારે આઠમા વિધાનનો અમલ કરી ઉપયોગકર્તાએ ઉમેરેલી તમામ સંખ્યાઓનો સરવાળો દર્શાવવામાં આવે છે.

નેસ્ટેડ લૂપ (Nested loop)

લૂપ નિયંત્રણના પ્રકારને અનુસાર એક લૂપ નિયંત્રણનો અન્ય લૂપની અંદર ઉપયોગ કરી જુદા જુદા પ્રકારનું નેસ્ટિંગ શક્ય છે. ઉદાહરણ તરીકે *for* લૂપની અંદર *while* કે *do...while* લૂપનો ઉપયોગ કરી શકાય છે. જેમ કે, 1થી 100 વચ્ચેની અવિભાજ્ય સંખ્યાઓ દર્શાવવા માટે *for* લૂપનો ઉપયોગ કરી સંખ્યાઓનું લૂપ લખી શકાય છે અને આ લૂપની અંદર *while* લૂપનો ઉપયોગ કરી આપેલ સંખ્યા અવિભાજ્ય છે કે નહીં તે શોધી શકાય છે.

યોગ્ય લૂપની પસંદગી (Selecting a loop)

અત્યાર સુધીમાં આપણે આ પ્રકરણમાં સી ભાષાના ત્રણ લૂપિંગ બંધારણો - *for*, *while* અને *do...while* ની ચર્ચા કરી. પ્રોગ્રામમાં વધુ યોગ્ય લૂપની પસંદગી કરવા માટે નીચે આપેલા મુદ્દાઓનો ઉપયોગ કરી શકાય :

- આપેલ પ્રશ્ન માટે પ્રવેશ-નિયંત્રિત (entry controlled) કે નિર્ગમ-નિયંત્રિત (exit controlled) લૂપની જરૂર પડશે તે નક્કી કરો.
- પ્રવેશ-નિયંત્રિત લૂપ માટે *for* કે *while* લૂપનો ઉપયોગ કરવો જોઈએ. ગણતરી આધારિત નિર્ગમન શરત (counter based exit condition) માટે *for* લૂપનો ઉપયોગ સલાહભર્યો છે.
- નિર્ગમ-નિયંત્રિત લૂપ માટે *do...while* બંધારણનો ઉપયોગ કરવો જોઈએ.

આ ઉપરાંત, પ્રોગ્રામમાં કોઈ પણ લૂપનો ઉપયોગ કરતી વખતે નીચેના ત્રણ મુદ્દાઓનું ધ્યાન રાખવાથી લૂપનું અનિચ્છનીય વર્તન અટકાવી શકાય છે :

- લૂપમાં ગણતરીનો આરંભ કરવો.
- લૂપમાંથી બહાર આવવા માટેની નિર્ગમન શરત (exit condition) ચકાસવી.
- લૂપમાં ગણતરી માટેના કાઉન્ટરની કિંમત વધારવી અથવા ઘટાડવી.

લૂપનો કોઈ ભાગ છોડી દેવો (Skipping a part of loop)

પ્રોગ્રામના અમલીકરણ વખતે ઘણીવાર આપણે કેટલાંક વિધાનો અથવા ફેરા (iterations)ને છોડી દેવા માંગતા હોઈએ છીએ. સી ભાષામાં નીચેનાં વિધાનોનો ઉપયોગ કરવાથી આ શક્ય છે :

- break વિધાન
- continue વિધાન

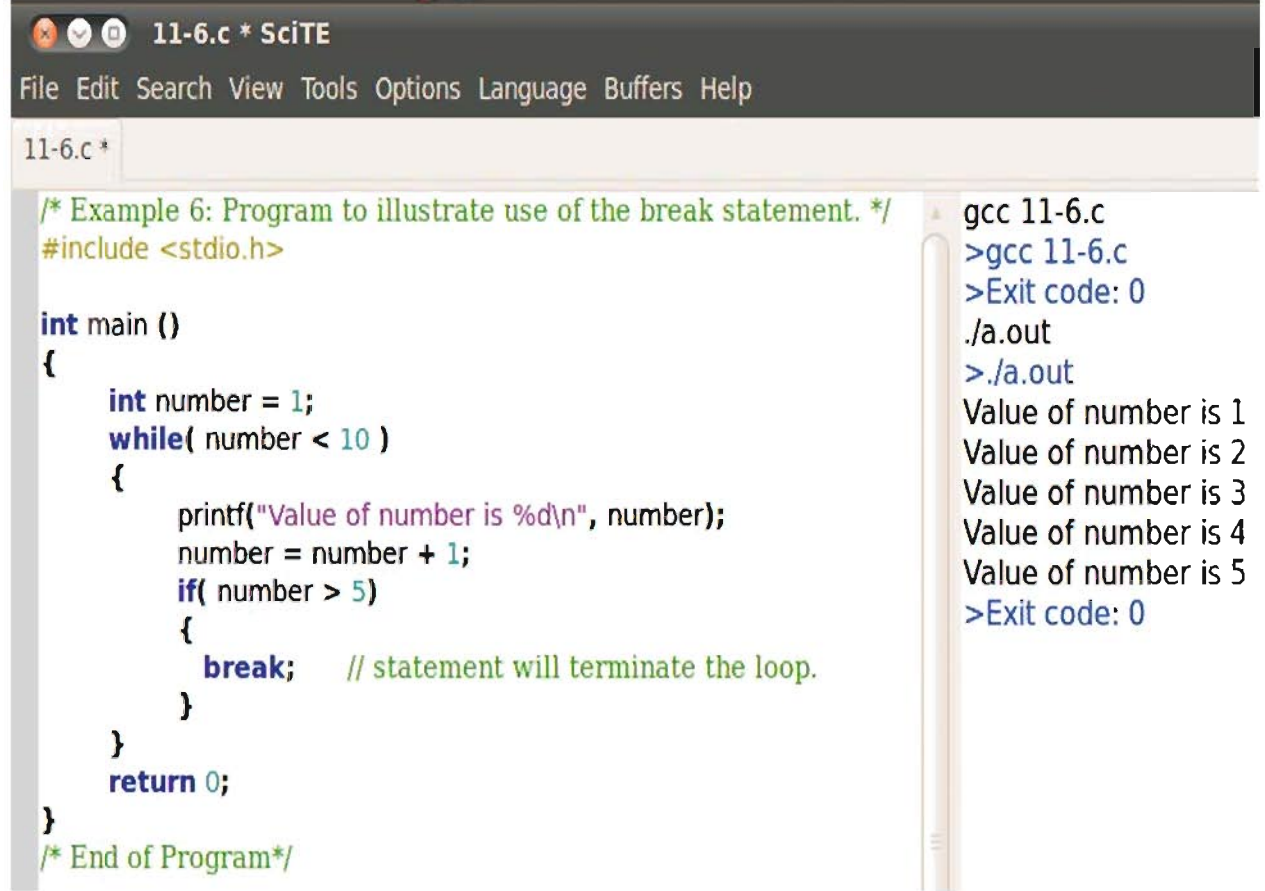
break વિધાન (The break statement)

લૂપના અમલ દરમિયાન કેટલીકવાર લૂપનાં અન્ય વિધાનોનો અમલ કર્યા વિના લૂપને અચાનક જ અટકાવી દેવાની જરૂર પડતી હોય છે. *break* વિધાનના ઉપયોગથી આ શક્ય છે. સી પ્રોગ્રામિંગ ભાષામાં *break* વિધાનના ઉપયોગની માહિતી નીચે જણાવેલ છે :

- લૂપમાં *break* વિધાન આવે ત્યારે લૂપને તત્કાલ અટકાવવામાં આવે છે તથા પ્રોગ્રામ નિયંત્રણ દ્વારા લૂપ પછીના વિધાનનો અમલ કરવામાં આવે છે.
- *switch* વિધાનમાં *case*નો અમલ અટકાવવા માટે પણ *break* વિધાનનો ઉપયોગ કરવામાં આવે છે.

નેસ્ટેડ લૂપના કિસ્સામાં *break* વિધાન તત્કાલ અમલમાં હોય તે લૂપનું અમલીકરણ અટકાવે છે અને કોડના જૂથ પછી આવેલા વિધાનનો અમલ શરૂ કરવામાં આવે છે.

પ્રોગ્રામમાં *break* વિધાનનો ઉપયોગ ઉદાહરણ 14.6 દ્વારા દર્શાવ્યો છે. આકૃતિ 14.13માં ઉદાહરણ 14.6નું કોડ-લિસ્ટિંગ તથા પરિણામ દર્શાવ્યું છે.



The screenshot shows a code editor window titled "11-6.c * SciTE". The menu bar includes File, Edit, Search, View, Tools, Options, Language, Buffers, and Help. The code in the editor is as follows:

```

/* Example 6: Program to illustrate use of the break statement. */
#include <stdio.h>

int main ()
{
    int number = 1;
    while( number < 10 )
    {
        printf("Value of number is %d\n", number);
        number = number + 1;
        if( number > 5)
        {
            break;    // statement will terminate the loop.
        }
    }
    return 0;
}
/* End of Program*/

```

On the right side, the terminal output is shown:

```

gcc 11-6.c
>gcc 11-6.c
>Exit code: 0
./a.out
>./a.out
Value of number is 1
Value of number is 2
Value of number is 3
Value of number is 4
Value of number is 5
>Exit code: 0

```

આકૃતિ 14.13 : ઉદાહરણ 14.6નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી

main() વિધેય પછીનું પ્રથમ વિધાન એક પૂર્ણાંક ચલની પ્રારંભિક કિંમત નિશ્ચિત કરે છે. લૂપને નવ (9) વખત ફેરવવા માટે *while* વિધાનમાં શરત લખવામાં આવી છે. લૂપની અંદર આવેલું *printf* વિધાન *number* ચલની કિંમત દર્શાવશે. *number* ચલની કિંમત દર્શાવ્યા પછી તેની કિંમતમાં 1નો વધારો કરવામાં આવશે. ત્યારપછી, *if* વિધાનનો અમલ કરવામાં આવશે. પરંતુ પ્રોગ્રામના અમલ દરમિયાન જ્યારે *if* વિધાનનું મૂલ્યાંકન *true* મળશે ત્યારે *break* વિધાનનો અમલ કરવામાં આવશે, જે *while* લૂપનો અમલ અટકાવશે. આમ, આ પ્રોગ્રામમાં *while* લૂપમાં આવેલું *printf* વિધાન માત્ર પાંચ વખત અમલમાં મુકાશે.

continue વિધાન (The continue statement)

continue વિધાન કેટલેક અંશે *break* વિધાન જેવું કાર્ય કરે છે. જો કે, લૂપને અટકાવવાને બદલે *continue* વિધાન લૂપના અન્ય કોડનું અમલીકરણ સ્થગિત કરી લૂપના ત્યાર પછીના ફેરાનો અમલ શરૂ કરે છે.

ઉદાહરણ તરીકે, *for* લૂપમાં *continue* વિધાન દ્વારા શરતની ચકાસણી અને ચલમાં કરવામાં આવતા વધારાના વિભાગનો અમલ કરવામાં આવે છે. *while* અને *do...while* લૂપ માટે *continue* વિધાન પ્રોગ્રામના નિયંત્રણને શરતની ચકાસણી તરફ મોકલે છે.

પ્રોગ્રામમાં *continue* વિધાનનો ઉપયોગ ઉદાહરણ 14.7માં દર્શાવ્યો છે. આકૃતિ 14.14માં ઉદાહરણ 14.7નું કોડ-લિસ્ટિંગ અને પરિણામ દર્શાવ્યાં છે.

```

11-7.c
/* Example 7: Program to illustrate the use of continue statement. */
#include <stdio.h>

int main ()
{
    int number = 1;
    printf("Value of number is %d\n", number);

    while(number < 6)
    {
        number = number + 1;
        if( number == 3)
        {
            continue; // statement skip the current iteration of the loop
        }
        printf("Value of number is %d\n", number);
    }
    return 0;
}
/* End of Program*/

```

```

gcc 11-7.c
>gcc 11-7.c
>Exit code: 0
./a.out
>./a.out
Value of number is 1
Value of number is 2
Value of number is 4
Value of number is 5
Value of number is 6
>Exit code: 0

```

આકૃતિ 14.14 : ઉદાહરણ 14.7નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી

પ્રથમ વિધાન દ્વારા પ્રોગ્રામમાં જરૂરી એવા number ચલની ઘોષણા કરી પ્રારંભની કિંમત આપવામાં આવે છે. બીજું વિધાન number ચલની કિંમત દર્શાવે છે. લૂપનો પાંચ વખત અમલ કરવા માટે while વિધાનમાં શરત લખવામાં આવી છે. while લૂપની અંદર number ચલની કિંમત 1 વડે વધારવામાં આવે છે. ત્યાર પછી if વિધાન શરતની ચકાસણી કરે છે. જ્યારે તેનું મૂલ્યાંકન true મળે ત્યારે continue વિધાનનો અમલ કરવામાં આવે છે, જે વર્તમાન ફેરાને અટકાવે છે. તેથી number ચલની કિંમત 3 દર્શાવતું નથી અને ત્યારપછીના ફેરાનો અમલ કરે છે. તેથી આ પ્રોગ્રામ number ચલની 4, 5 અને 6 સંખ્યા દર્શાવે છે.

અનંત લૂપ (The Infinite loop)

જો કોઈ લૂપ સતત ચાલ્યા જ કરે અને પ્રોગ્રામનું નિયંત્રણ ક્યારેય તેની બહાર આવી ન શકે તો તેવા લૂપને અનંત લૂપ કહે છે. લૂપના તર્કમાં નિર્ગમન શરત(exit condition)ની અનુપલબ્ધતાને કારણે લૂપ અનંત લૂપ બને છે. આકૃતિ 14.15માં આપવામાં આવેલા ઉદાહરણ 14.8ના પ્રોગ્રામનો તર્ક જુઓ.

```

11-8.c
/* Example 8: Program to illustrate infinite loop.*/
#include <stdio.h>

int main ()
{
    for( ; ; )
    {
        printf(" This for loop will run forever.....\n " );
    }
    return 0;
}
/* End of program*/

```

```

This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....
This for loop will run forever.....

```

આકૃતિ 14.15 : ઉદાહરણ 14.8નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી

જ્યારે આ કોડને કમ્પાઇલ કરી અમલ કરવામાં આવશે ત્યારે તેનું અમલીકરણ સતત અને અવિરત ચાલ્યા કરશે. સી પ્રોગ્રામના *for* લૂપમાં તમામ ત્રણ પદાવલિઓ મરજિયાત છે. જો *for* લૂપમાં શરત માટેની પદાવલિ લખવામાં ન આવે તો તેને *true* સમજી લેવામાં આવે છે. પ્રારંભિક કિંમત આપવા માટેની પદાવલિ તથા ચલની કિંમતમાં વધારો કરવા માટેની પદાવલિ *for* લૂપની અંદર હોઈ શકે છે, પરંતુ *for* (; ;) બંધારણનો ઉપયોગ કરવાથી સી પ્રોગ્રામનું અમલીકરણ અવિરત ચાલ્યા કરે છે. અનંત લૂપને અટકાવવા માટે *Ctrl + C* કીનો ઉપયોગ કરી શકાય છે.

સારાંશ

પ્રોગ્રામરને વિધાન કે વિધાનોના સમૂહનું અમલીકરણ એકથી વધુ વખત કરવા માટેની અનુમતિ આપતા સી ભાષાના **લૂપ નિયંત્રણ બંધારણ** વિશે આપણે આ પ્રકરણમાં અભ્યાસ કર્યો. સી ભાષા દ્વારા પૂરા પાડવામાં આવતા ત્રણ લૂપ નિયંત્રણ બંધારણ(*for*, *while* અને *do...while*)નો અભ્યાસ કર્યો. આપણા પ્રોગ્રામમાં પ્રવેશ નિયંત્રણ લૂપ અને નિર્ગમ નિયંત્રણ લૂપનો ઉપયોગ કેવી રીતે કરી શકાય તેનો પણ આપણે અભ્યાસ કર્યો. પ્રોગ્રામના અમલ દરમિયાન કેટલાંક વિધાનો કે ફેરાનું અમલીકરણ મોકૂફ રાખવા *break* અને *continue* વિધાનનો ઉપયોગ કેવી રીતે કરી શકાય તે આપણે આ પ્રકરણના અંતમાં શીખ્યા.

સ્વાધ્યાય

1. જરૂરી ઉદાહરણો સાથે *for* લૂપ બંધારણની વાક્યરચના સમજાવો.
2. લૂપિંગ વિધાનો એટલે શું? સી પ્રોગ્રામમાં લૂપિંગ વિધાનોનો ઉપયોગ જણાવો.
3. *while* અને *do...while* લૂપ વચ્ચેનો તફાવત જણાવો.
4. *break* અને *continue* વિધાન દ્વારા લૂપના કોઈ ભાગનું અમલીકરણ કેવી રીતે મોકૂફ રાખી શકાય તે સમજાવો.
5. પ્રવેશ નિયંત્રણ અને નિર્ગમ નિયંત્રણ લૂપ એટલે શું? યોગ્ય ઉદાહરણ આપો.
6. ખાલી જગ્યા પૂરો :
 - (a) લૂપમાં આવેલા નિયંત્રણ વિધાનના સ્થાનને આધારે લૂપને _____ અને _____ એમ બે વર્ગોમાં વર્ગીકૃત કરી શકાય.
 - (b) *for* લૂપ _____ નિયંત્રિત લૂપ છે.
 - (c) *do...while* લૂપ _____ નિયંત્રિત લૂપ છે.
 - (d) નિર્ગમ નિયંત્રિત લૂપમાં લૂપનાં વિધાનોનો ઓછામાં ઓછા _____ વખત અમલ કરવામાં આવે છે.
 - (e) લૂપ નિયંત્રણ બંધારણમાંથી તત્કાલ બહાર આવવા માટે _____ વિધાનનો ઉપયોગ કરી શકાય.
 - (f) લૂપમાં આવેલાં પછીનાં વિધાનોના અમલને અટકાવી ત્યાર પછીનો ફેરો શરૂ કરવા માટે _____ વિધાન જરૂરી છે.
7. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો :
 - (a) *for* લૂપની અંદર *do...while* લૂપનો ઉપયોગ કરી શકાતો નથી.
 - (b) લૂપમાં આવેલ *break* વિધાનનું અમલીકરણ પ્રોગ્રામનો અમલ તત્કાલ અટકાવે છે.

- (c) for લૂપના મથાળામાં પ્રારંભિક કિંમત ઉમેરવી, શરત લખવી અને ચલમાં વધારો કરવા માટેનો વિભાગ ઉમેરવો - આ તમામ વૈકલ્પિક છે.
- (d) switch વિધાનના caseને અટકાવવા માટે break વિધાનનો ઉપયોગ કરી શકાતો નથી.
- (e) પ્રારંભિક કિંમત ઉમેરવી, શરત અને ચલમાં વધારો કરવા માટેના વિભાગને અલ્પવિરામ દ્વારા છૂટા પાડવામાં આવે છે.
- (f) દરેક while વાક્યાંશ સાથે do હોવું ફરજિયાત છે.
- (g) દરેક do વાક્યાંશ સાથે તેને સંબંધિત while હોવું ફરજિયાત છે.

8. આપેલ વિકલ્પોમાંથી સાચો વિકલ્પ પસંદ કરો :

(1) નીચેના પ્રોગ્રામ-ખંડમાં printf વિધાનનો અમલ કેટલીવાર કરવામાં આવશે ?

```
int num1 = 3, num2 = 6;
while(num1 < num2) {
    printf("Hello students...");
    num1 = num1 + 1;
}
```

- (a) 1 (b) 2 (c) 3 (d) 6

(2) નીચેના પ્રોગ્રામ-ખંડનું પરિણામ શું મળશે ?

```
int a = 5, b = 10;
do
{
    a = a + 1;
```

```
}while(a <= b);
```

```
printf("%d", a);
```

- (a) 10 (b) 9 (c) 11 (d) 15

(3) નીચેના પ્રોગ્રામ-ખંડનું પરિણામ શું મળશે ?

```
int main( ){
int i;
for(i = 0; i <10; i++);
printf("%d", i);
}
```

- (a) 0123456789 (b) 9 (c) 10 (d) ભૂલનો સંદેશ

(4) નીચેના કોડના અમલ પછી number ચલની કિંમત કઈ હશે ?

```
int number = 1;
while(number < 5){
    number = number + 1;
    if( number == 3) {
        continue;
    }
}
```

(a) 1

(b) 3

(c) 5

(d) એક પણ નહીં

પ્રાયોગિક સ્વાધ્યાય

1. નીચેના for લૂપને તેના સમકક્ષ while લૂપમાં ફેરવો :

(a) for(number=0; number<5; number++) {
printf("%d", number);
}

(b) for(number=5; number >0; number —) {
printf("%d", number);
}

2. while લૂપમાં સંમિશ્રિત (compound) વિધાનોનો ઉપયોગ દર્શાવો.

3. for લૂપનો ઉપયોગ કરી પ્રથમ 100 પૂર્ણાંકોનો સરવાળો શોધવા માટેનો પ્રોગ્રામ લખો.

4. while લૂપનો ઉપયોગ કરી પ્રથમ 100 બેકી પૂર્ણાંક સંખ્યાઓનો સરવાળો કરવા માટેનો પ્રોગ્રામ બનાવો.

5. આપેલ લીટીઓની સંખ્યા માટે નીચેની પેટર્ન દર્શાવવા માટેનો પ્રોગ્રામ લખો. ઉદાહરણ તરીકે જો લીટીની સંખ્યા 4 આપવામાં આવે તો નીચે દર્શાવેલ પરિણામ દર્શાવવામાં આવે.

```
*
* *
* * *
* * * *
* * *
* *
*
```

6. સ્ક્રીન પર નીચે દર્શાવેલ શ્રેણી દર્શાવવા માટેનો for લૂપ પ્રોગ્રામ લખો :

(a) 1, 3, 5, 7, 9, 11

(b) 1, 2, 4, 8, 16, 32, 64

(c) 10, 8, 6, 4, 2, 0

(d) -10, -8, -6, -4, -2, 0

7. અપર કેસ અને લોઅર કેસ મૂળાક્ષરોની આસ્કી (ASCII) કિંમતો દર્શાવવા માટેનો પ્રોગ્રામ લખો.

8. ઉપયોગકર્તા 'Z' દાખલ ન કરે ત્યાં સુધી આપવામાં આવેલા તમામ અક્ષરોની સંખ્યા ગણવા તથા દર્શાવવા માટેનો પ્રોગ્રામ લખો.



એરે

‘સી’ પ્રોગ્રામિંગમાં અત્યાર સુધી આપણે મૂળભૂત ડેટા પ્રકાર જેવા કે int, float, char, double, વગેરેનો ઉપયોગ કર્યો. આ બધા ચલના પ્રકારો એક સમયે માત્ર એક જ કિંમત સાચવી શકે છે. એ કિંમત ઉપર પ્રક્રિયા કરવા માટે આપણને એક ચલની જરૂર પડે છે. તેથી જો આપણે પાંચ કિંમતો પર પ્રક્રિયા કરવી હોય તો એ માટે પાંચ જુદા જુદા ચલની જરૂર પડે. તો હવે એક એવા પ્રોગ્રામની કલ્પના કરો કે, જેમાં આપણને અનેક ડેટા પર પ્રક્રિયા કરવી પડતી હોય. આવી પરિસ્થિતિમાં, C ભાષા એક ખાસ પ્રકારની એરે ડેટા ટાઇપ આપે છે. આ પ્રકરણમાં આપણે એ શીખીશું કે C ભાષામાં ઉપલબ્ધ જુદા જુદા પ્રકારના એરેનો કેવી રીતે ઉપયોગ કરવો.

એરેની જરૂરિયાત (Need of arrays)

કલ્પના કરો કે, નીચે મુજબની જીવનની કેટલીક વાસ્તવિક પરિસ્થિતિ માટે આપણે C પ્રોગ્રામ લખવા છે :

- વર્ગના બધા વિદ્યાર્થીઓના ગુણ સાચવવા
- સુપર માર્કેટની બધી ચીજવસ્તુઓની કિંમત સાચવવી
- કર્મચારીઓ અને તેમના સંપર્ક સૂત્રની યાદી
- વર્ગના દરેક વિદ્યાર્થીની ઉંચાઈ અને વજનની માહિતી સાચવવી

દરેક વિદ્યાર્થીઓના ગુણની માહિતીનું વ્યવસ્થાપન કરવા marks0, marks1, marks2, marks3, marks4 અને marks59, એમ 60 વિદ્યાર્થીઓના ગુણ માટે 60 જુદા જુદા ચલ બનાવવાને બદલે આપણે એક જ એરે ચલ જેમ કે, marks[60] વ્યાખ્યાયિત કરવી શકાય. હવે આ એરેમાંથી અલગ અલગ વિદ્યાર્થીઓના ગુણ જાણવા માટે આપણે marks[0], marks[1], marks[2] ... એ પ્રમાણે ચલનો ઉપયોગ કરી શકીએ. કમ્પ્યુટરની સ્મૃતિ(મેમરી)માં marks[60] નામનો એરે આકૃતિ 15.1માં દર્શાવ્યા મુજબ દેખાશે.

First element	Second element	Third element	Last element
marks[0]	marks[1]	marks[2]	marks[3]	marks[59]

આકૃતિ 15.1 : સ્મૃતિમાં એરેના માળખાનો નમૂનો

એરેને સંબંધિત એવા નીચે મુજબના અગત્યના મુદ્દા નોંધો :

- તે એક જ ડેટાપ્રકારના ઘટકોનો સમૂહ છે.
- તે નિયત કદ ધરાવતા ઘટકોનો ક્રમિક સમૂહ છે. એરે સળંગ સ્મૃતિસ્થાનો રોકે છે.
- એરેનો કોઈ પણ ઘટક તેના અનુક્રમ દ્વારા ઉપયોગમાં લઈ શકાય છે. એરેના નામની પાછળ ચોરસ કૌંસમાં દર્શાવતા અનુક્રમ(Index number)ને ‘સબસ્ક્રિપ્ટ’ (subscript) તરીકે પણ ઓળખવામાં આવે છે.
- આ સબસ્ક્રિપ્ટ તરીકે કોઈ પણ પૂર્ણાંક અથવા પૂર્ણાંક પદાવલિ જ હોવી જોઈએ.
- સબસ્ક્રિપ્ટનો ક્રમ શૂન્યથી શરૂ થાય છે.

એરેના પ્રકારો (Types of arrays)

‘સી’ ભાષામાં એરેને નીચે મુજબ બે પ્રકારોમાં વિભાજિત કરી શકાય :

- એક પરિમાણીય એરે (Single or One dimensional array)
- બહુપરિમાણીય એરે (Multi dimensional array)

એક પરિમાણીય એરેમાં એક ઊભી હરોળ અથવા એક આડી હરોળ હોય છે જ્યારે બહુપરિમાણીય એરેમાં એક કે વધુ આડી હરોળ અથવા એક કે વધુ બહુ ઊભી હરોળ હોય છે. ઊભી અને આડી હરોળની સંખ્યા પ્રોગ્રામની જરૂરિયાત અનુસાર ઉપયોગકર્તા દ્વારા નક્કી કરવામાં આવે છે. આપણે માત્ર C ભાષા દ્વારા આપવામાં આવતા એક પરિમાણીય અને દ્વિપરિમાણીય એરેની જ ચર્ચા કરીશું.

એકપરિમાણીય એરે વ્યાખ્યાયિત કરવા (Declaration of single dimensional array)

C ભાષામાં એરેને વ્યાખ્યાયિત કરવા સ્મૃતિ ઘટકોમાં આપણે જે પ્રકારનો ડેટા સાચવવા માંગતા હોઈએ તે અને એરેમાં કેટલા ઘટકો રાખવા છે તેને ધ્યાને રાખી આપણે નીચે મુજબ એરે વ્યાખ્યાયિત કરીએ છીએ :

datatype arrayname[size];

- **datatype:** એરે કયા પ્રકારના ઘટકો સાચવે તે નક્કી કરે છે. જો આપણે એરેમાં શાબ્દિક માહિતી (characters) સાચવવા ઇચ્છતા હોઈએ તો એરેનો પ્રકાર 'char' હોય. એ જ રીતે, પૂર્ણાંક (integers) સંખ્યા સાચવવી હોય તો એરેનો પ્રકાર 'int' હોય. ટૂંકમાં, datatype તરીકે C ભાષાનો કોઈ પણ માન્ય ડેટાપ્રકાર હોઈ શકે.
- **arrayname:** આ પ્રોગ્રામ દ્વારા એરેને આપવામાં આવતું નામ છે. આ નામ તરીકે કોઈ પણ અક્ષરોનો સમૂહ ચાલી શકે. પરંતુ એવી સલાહ આપવામાં આવે છે કે, એરેને એક અર્થસભર નામ આપવું જોઈએ. એરેનું નામ બને ત્યાં સુધી પ્રોગ્રામ દ્વારા એરેમાં શું સાચવવામાં આવે છે તેના સંદર્ભમાં હોવું જોઈએ.
- **size:** size તરીકે આપવામાં આવતી કિંમત દ્વારા જ એરેમાં કેટલા ઘટકો સાચવી શકાશે તે નક્કી થાય છે. આ size તરીકે શૂન્યથી મોટો કોઈ પણ પૂર્ણાંક અચલ હોવો આવશ્યક છે.

ઉદાહરણ તરીકે, 60 વિદ્યાર્થીઓના ગુણની વિગતો સાચવવા આપણે નીચે મુજબનું વિધાન લખી શકીએ :

int marks[60];

અહીં એ નોંધ લો કે, વ્યાખ્યાયિત કરાયેલો marks નામનો એરે માત્ર 60 પૂર્ણાંક સંખ્યાઓ સંગ્રહવા માટે જ સક્ષમ છે. શાબ્દિક માહિતી, અપૂર્ણાંક સંખ્યાઓ અને વધુ મોટી કિંમતો (double) સાચવી શકે તેવા એરેના ઉદાહરણ નીચે વ્યાખ્યાયિત કરેલ છે :

char string [20];

float percentages [20];

double numbers [20];

એકપરિમાણીય એરેને કિંમતો આપવી (Assigning values to single dimensional arrays)

કોઈ પણ એરેને પ્રારંભિક કિંમતો આપવા માટે C ભાષામાં બે રીતો છે :

- (i) કંપાઈલેશન સમયે એરેને પ્રારંભિક કિંમતો આપવી.
- (ii) પ્રોગ્રામ ચાલે ત્યારે એરેને પ્રારંભિક કિંમતો આપવી.

કંપાઈલેશન સમયે એરેને પ્રારંભિક કિંમતો આપવી (Compile time array initialization)

એરેને વ્યાખ્યાયિત કરતી વખતે કોઈ સામાન્ય ચલની જેમ જ કિંમતો આપી શકાય છે. વિવિધ એરે ઘટકોને કિંમતો આપવા માટેની સામાન્ય વાક્યરચના નીચે મુજબ છે :

datatype arrayname[size] = {value1, value2, , valueN};

અહીં, value1, value2, , valueN એ એરેના એક પછી એક આગળના ઘટકો માટે પ્રારંભિક કિંમત પૂરી પાડે છે. નીચે કેટલાંક ઉદાહરણ આપ્યાં છે જેમાં આપણે એરેના તમામ ઘટકોનું કદ નક્કી કરી તેને કિંમતો આપી શકીએ છીએ.

int marks[5] = {60, 65, 70, 75, 80};

આ ઉદાહરણ marks નામનો એરે ચલ વ્યાખ્યાયિત કરશે, જે પાંચ ઘટકો સાચવી શકશે. આ ઘટકોની કિંમત

marks[0] = 60, marks[1] = 65, marks[2] = 70, marks[3] = 75 અને marks[4] = 80 ગોઠવાશે. આવી રીતે એરે વ્યાખ્યાયિત કરવાના કિસ્સામાં જો એરેના કદ કરતાં કિંમતોની યાદી નાની હશે, તો શરૂઆતના એક પછી એક ઘટકોને કિંમત આપવામાં આવશે અને પાછળના ઘટકોની કિંમત શૂન્ય રાખવામાં આવશે. ઉદાહરણ તરીકે,

```
int marks[5] = {60, 65, 70};
```

પ્રથમ ત્રણ એરે ઘટકોની પ્રારંભિક કિંમત marks[0] = 60, marks[1] = 65 અને marks[2] = 70 થશે. જ્યારે એ પછીના marks[3] અને marks[4] ઘટકની પ્રારંભિક કિંમત 0 અપાશે.

જો એરેને વ્યાખ્યાયિત કરતી વખતે જ પ્રારંભિક કિંમત આપવામાં આવે તો એરેનું કદ દર્શાવવું મરજિયાત છે. જો આપણે એરેનું કદ નહીં દર્શાવીએ તો કંપાઇલર આપમેળે છગડિયા કૌંસ({})ની અંદર આપેલી કિંમતોની સંખ્યા ગણીને નક્કી કરી લેશે. ઉદાહરણ તરીકે, નીચેનું વિધાન

```
int marks[ ] = {60, 65, 70, 75, 80};
```

પાંચ ઘટકો ધરાવતો marks નામનો એરે વ્યાખ્યાયિત કરશે જેને છગડિયા કૌંસમાં આપેલી કિંમતો દ્વારા પ્રારંભિક કિંમતો આપવામાં આવશે.

તો ચાલો, ઉદાહરણ 15.1 સાથે એક પરિમાણીય એરેને કંપાઇલેશન સમયે પ્રારંભિક કિંમત આપવાના ખ્યાલનો અભ્યાસ કરીએ. આકૃતિ 15.2 ઉદાહરણ 15.1નું કોડ-લિસ્ટિંગ અને પરિણામ રજૂ કરે છે.

```

12-1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

12-1.c
/* Example 1: Program to illustrate compile time initialization of
one dimensional array.*/

#include<stdio.h>
int main()
{
    int number[5] = {10,20,30,40,50}; // Compile time initialization
    int i;
    for( i = 0; i < 5; i++)           // Loop to display array elements
    {
        printf(" %d \n", number[i] );
    }
    return 0;
}
/* End of Program*/

gcc 12-1.c
>gcc 12-1.c
>Exit code: 0
./a.out
>./a.out
10
20
30
40
50
>Exit code: 0

```

આકૃતિ 15.2 : ઉદાહરણ 15.1નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી

main વિધેય પછીની પ્રથમ લીટીમાં number નામનો એરે વ્યાખ્યાયિત કરવામાં આવ્યો છે અને કંપાઇલેશન સમયે જ તમામ એરે ઘટકોને છગડિયા કૌંસમાં આપેલી કિંમતો વડે પ્રારંભિક કિંમત આપવામાં આવે છે. for લૂપ number નામના એરેના અલગ-અલગ ઘટકને દર્શાવે છે.

અમલ દરમ્યાન એરેને પ્રારંભિક કિંમતો આપવી

જ્યારે આપણને પ્રોગ્રામનો અમલ કરતી વખતે ઉપયોગકર્તા પાસેથી કિંમતો મેળવવાની જરૂર પડતી હોય, તો આપણે અમલ દરમ્યાન એરેની પ્રારંભિક કિંમત આપવાની પદ્ધતિનો ઉપયોગ કરી શકીએ. એરેમાં સંગ્રહિત કિંમતોનો કેવી રીતે ઉપયોગ કરવો તે પણ આ પ્રોગ્રામ દર્શાવે છે. આકૃતિ 15.3, ઉદાહરણ 15.2નું કોડ-લિસ્ટિંગ આપે છે, જ્યારે આકૃતિ 15.4 તેનું પરિણામ દર્શાવે છે.

```
12-2.c * SciTE
File Edit Search View Tools Options Language Buffers Help

12-2.c *
/* Example 2: Program to illustrate runtime array initialization and accessing values
from one dimensional array. Program will read marks of five students and display the
average marks.*/

#include<stdio.h>
int main()
{
    int marks[5], i, total=0;
    for( i = 0; i < 5; i++)           // loop to read marks of five students
    {
        printf("Enter marks[%d] :", i);
        scanf("%d", &marks[i]);
    }
    for( i = 0; i < 5; i++)           // loop to find total marks
    {
        total = total + marks[i];
    }
    printf("\nThe average marks of students are %d. \n", total/5);
    return 0;
}
/* End of Program*/
```

આકૃતિ 15.3 : ઉદાહરણ 15.2નું કોડ-વિસ્દેશ

```
kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 12-2.c
kpp@ubuntu:~$ ./a.out
Enter marks[0] :5
Enter marks[1] :10
Enter marks[2] :15
Enter marks[3] :20
Enter marks[4] :25

The average marks of students are 15.
kpp@ubuntu:~$
```

આકૃતિ 15.4 : ઉદાહરણ 15.2નું પરિણામ

સમજૂતી

પ્રોગ્રામની પ્રથમ *for* લૂપ પાંચ વિદ્યાર્થીઓના ગુણ વાંચશે. બીજી *for* લૂપ *marks* નામના એરેના અલગ-અલગ ઘટક વાંચીને કુલ ગુણ ગણશે. છેલ્લું *printf* વિધેય વિદ્યાર્થીઓના સરેરાશ ગુણ દર્શાવશે.

કેરેક્ટર એરે તરીકે શબ્દમાળા

શબ્દમાળા એ અક્ષરોની હારમાળા છે, જે સામાન્ય રીતે એક ઘટક તરીકે લેવામાં આવે છે. C ભાષા ડેટાના એક પ્રકાર તરીકે શબ્દમાળા માટે *string* ડેટા પ્રકારને સમર્થન આપતી નથી. જો કે, તેમ છતાં તે આપણને કેરેક્ટર એરે

તરીકે કોઈ પણ શાબ્દિક માહિતીને રજૂ કરવાની છૂટ આપે છે. શાબ્દિક ચલ એ C ભાષાનું કોઈ પણ માન્ય ચલનું નામ છે, અને તે હંમેશાં કેરેક્ટર એરે તરીકે જ વ્યાખ્યાયિત કરવામાં આવે છે. શાબ્દિક ચલને વ્યાખ્યાયિત કરવા માટેનું સામાન્ય સ્વરૂપ નીચે મુજબ છે:

char stringname [size];

અહીં, size એ stringname નામના ચલમાં અક્ષરોની સંખ્યા દર્શાવે છે. નીચે કેટલાક માન્ય શાબ્દિક ચલને વ્યાખ્યાયિત કરવાનાં ઉદાહરણ આપેલ છે :

char student_name [20];

char city [50];

char state [20];

અહીં એ નોંધવું જરૂરી છે કે C ભાષામાં શાબ્દિક માહિતીનો અંત ખાલી અક્ષર (null character) તરીકે ઓળખાતા '\0'; ખાસ ચિહ્નથી આવે છે. આ ખાસ ચિહ્ન પ્રોગ્રામને શાબ્દિક માહિતીનો અંત ઓળખવામાં મદદરૂપ બને છે. આમ, કોઈ પણ શાબ્દિક માહિતીના અંતે '\0' (null character) ચિહ્ન મૂકવું જરૂરી હોવાથી, આપણી શાબ્દિક માહિતીના બધા શબ્દોને સમાવી શકાય તે માટે કેરેક્ટર એરેમાં શબ્દોની કુલ સંખ્યા ઉપરાંત એક વધારાના સંગ્રહસ્થાનની જરૂર પડે છે.

સાંખ્યિક એરેની જેમ જ, શાબ્દિક એરેને પણ કંપાઈલ કરતી વખતે અથવા પ્રોગ્રામના અમલ દરમ્યાન એમ બે રીતે પ્રારંભિક કિંમતો આપી શકાય છે. કેરેક્ટર એરેને નીચે મુજબ જુદી જુદી રીતે પ્રારંભિક કિંમતો આપી શકાય :

char student_name[6] = "PURVA";

char student_name[6] = {'P', 'U', 'R', 'V', 'A', '\0'};

student_name નામના એરે કદ 6 ઘટકોનું રાખવા પાછળનું કારણ એ છે કે, PURVA શબ્દમાં પાંચ અક્ષર હોવાથી તેના 5 અને છેલ્લે null character ('\0') મૂકવા એક વધારાનો ઘટક એમ કુલ મળી 6 ઘટક. અહીં એ નોંધવું જોઈએ કે, જ્યારે તેના ઘટકોની યાદી રજૂ કરીને આપણે કેરેક્ટર એરેને પ્રારંભિક કિંમતો આપીએ ત્યારે શાબ્દિક માહિતીના અંતે null character '\0' દર્શાવવું જરૂરી છે. આપણે કદ દર્શાવ્યા વગર પણ નીચે મુજબ કેરેક્ટર એરે વ્યાખ્યાયિત કરી શકીએ :

char state[] = {'G', 'U', 'J', 'A', 'R', 'A', 'T', '\0'};

આ કિસ્સામાં state નામનો કેરેક્ટર એરે વ્યાખ્યાયિત થઈ જશે અને તેને 8 પ્રારંભિક કિંમતો અપાઈ જશે. નીચેનાં વિધેયોનો ઉપયોગ કરીને આપણે અગાઉ ચર્ચા કરેલ કેરેક્ટર એરેની માહિતી જોઈ શકીએ :

printf("The name of student is %s", student_name);

printf("The name of state is %s", state);

printf("The first character of your name is %c", student_name[0]);

બહુપરિમાણીય એરે (Multi dimensional array)

એકપરિમાણીય એરે માહિતીને રૈખિક ક્રમમાં જાળવે છે. જો કે, કેટલીક ઉપયોગી પદ્ધતિઓ (જેવી કે, સાંખ્યિક છબી, ચેસ રમતનું બોર્ડ વગેરે) સાથે સંકળાયેલ માહિતી દ્વિપરિમાણીય હોય છે. પ્રોગ્રામમાં આ પ્રકારની પદ્ધતિનો વિચાર કરતી વખતે આપણને બહુપરિમાણીય એરે બનાવવાની છૂટ આપે છે. અગાઉ જણાવ્યા મુજબ, બહુપરિમાણીય એરેમાં એક કરતાં વધુ ઊભી અને આડી હરોળ હોય છે. C ભાષામાં આપણે દ્વિપરિમાણીય, ત્રિપરિમાણીય અથવા N-પરિમાણીય એરે બનાવી શકીએ. બહુપરિમાણીય એરેનું સૌથી સાદું સ્વરૂપ દ્વિપરિમાણીય એરે છે. દ્વિપરિમાણીય એરેમાં આડી હરોળ માટે એક અને ઊભી હરોળ માટે એક એમ બે સબસ્ક્રિપ્ટ હોય છે. આ પ્રકરણમાં આપણે માત્ર દ્વિપરિમાણીય એરેની જ ચર્ચા કરીશું.

દ્વિપરિમાણીય એરેને વ્યાખ્યાયિત કરવો

અત્યાર સુધી આપણે એકપરિમાણીય એરેની ચર્ચા કરી, જે તેની અંદર ઘણી કિંમતો સાચવી શકે. જીવનમાં એવી ઘણી પરિસ્થિતિ હોય છે કે જેમાં આપણને ડેટાનું કોષ્ટક સાચવવાની જરૂર પડે. કોષ્ટક 15.1માં આપેલ કિસ્સા અંગે વિચારો. તે સોમવારથી શુક્રવાર દરમિયાન જુદા જુદા ત્રણ સેલ્સમેન દ્વારા વેચવામાં આવેલ વસ્તુની વિગત દર્શાવે છે.

	Monday	Tuesday	Wednesday	Thursday	Friday
Salesman1	100	150	200	250	200
Salesman2	200	250	300	350	300
Salesman3	150	200	250	300	250

કોષ્ટક 15.1 : વેચાણની વિગતો

આ કોષ્ટક જુદી જુદી 15 કિંમતો ધરાવે છે. અહીં એકપરિમાણીય એરે વ્યાખ્યાયિત કરવો સલાહભર્યો નથી. દ્વિપરિમાણીય એરેનો ઉપયોગ કરવો એ વધુ સારો વિકલ્પ છે. C ભાષાનો દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કરી કોષ્ટક 15.1ને સરળતાથી રજૂ કરી શકાય. દ્વિપરિમાણીય એરે નીચે મુજબ વ્યાખ્યાયિત કરી શકાય :

datatype arrayname [row size][column size];

row size અને *column size* શૂન્ય કરતાં મોટી પૂર્ણાંક અચલ સંખ્યા જ હોવી જોઈએ અને *datatype* તરીકે C ભાષાનો કોઈ પણ માન્ય ડેટા પ્રકાર હોઈ શકે. અહીં એ નોંધ કરો કે, C ભાષામાં બહુપરિમાણીય એરે એ આડી હાર અગ્રિમતા (row major) ધરાવે છે. બીજા શબ્દોમાં કહીએ તો, એરેને વ્યાખ્યાયિત કરતી વખતે દર્શાવવામાં આવતા બે કૌંસમાંથી પ્રથમ કૌંસ આડી હાર(rows)ની સંખ્યા નક્કી કરે છે.

કોષ્ટક 15.1માં દર્શાવેલ વિગતોને સાચવવા આપણે નીચે મુજબ દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કરી શકીએ છીએ:

int sales[3][5];

અહીં, આડી હારની સંખ્યા 3 છે, જે સેલ્સમેનને સંબોધે છે. (Salesman1 = 0, Salesman2 = 1 અને Salesman3 = 2) જ્યારે ઊભી હારોળની સંખ્યા 5 છે, જે અઠવાડિયાના જુદા જુદા વારને સંબોધવા ઉપયોગમાં લેવાશે. (Monday = 0, Tuesday = 1, Wednesday = 2, Thursday = 3 અને Friday = 4) આપણે એવું અનુમાન કર્યું છે કે, વેચાણના એકમોની સંખ્યા પૂર્ણાંક જ હશે. આકૃતિ 15.5 વેચાણની વિગતોની સ્મૃતિરચનાનો નમૂનો દર્શાવે છે.

Column index that refers to days					
	0	1	2	3	4
0	100	150	200	250	200
1	200	250	300	350	300
2	150	200	250	300	250

આકૃતિ 15.5 : વેચાણની વિગતોની સ્મૃતિરચના

હવે, Salesman1 દ્વારા Mondayના રોજ કરવામાં આવેલ ચીજવસ્તુના વેચાણની વિગત ચકાસવા, આપણે એરે ઘટક sales[0][0]નો ઉપયોગ કરી શકીએ. sales[0][0] ખાતે સાચવેલ કિંમત 100 છે. એ જ રીતે, Mondayના રોજ Salesman2 અને Salesman3 દ્વારા કરવામાં આવેલ વેચાણની વિગત ચકાસવા આપણે અનુક્રમે એરે ઘટક sales[1][0] અને sales[2][0]નો ઉપયોગ કરી શકીએ. આકૃતિ 15.5માં દર્શાવ્યા મુજબ sales[0][4] સ્થાન પર સાચવેલી કિંમત 200 છે અને તે Fridayના રોજ Salesman1 દ્વારા કરવામાં આવેલ વસ્તુનું વેચાણ દર્શાવે છે.

દ્વિપરિમાણીય એરેને પ્રારંભિક ક્રિમતો આપવી

એકપરિમાણીય એરેની જેમ જ, દ્વિપરિમાણીય એરેને પણ આપણે કંપાઈલ કરતી વખતે અથવા પ્રોગ્રામના અમલ દરમિયાન પ્રારંભિક ક્રિમતો આપી શકીએ.

કંપાઈલ કરતી વખતે દ્વિપરિમાણીય એરેને પ્રારંભિક ક્રિમતો આપવી

કોષ્ટકના પ્રોગ્રામને કંપાઈલ કરતી વખતે કોષ્ટક 15.1 માં દર્શાવેલ એરેને પ્રારંભિક ક્રિમતો આપવાનું કાર્ય નીચે મુજબ કરી શકાય :

```
int sales[3][5]={
    {100, 150, 200, 250, 200},
    {200, 250, 300, 350, 300},
    {150, 200, 250, 300, 250}
};
```

બહારના છગડિયા કૌંસની અંદર રહેલા છગડિયા કૌંસ, જે નિશ્ચિત સેલ્સમેન માટેની આડી હરોળ દર્શાવે છે, તે મરજિયાત છે. નીચે મુજબ પ્રારંભિક ક્રિમતો આપવાની પદ્ધતિ અગાઉની પદ્ધતિ જેવું જ સમાન કાર્ય કરે છે.

```
int sales[3][5]={100, 150, 200, 250, 200, 200, 250, 300, 350, 300, 150, 200, 250, 300, 250};
```

પ્રોગ્રામના અમલ વખતે દ્વિપરિમાણીય એરેને પ્રારંભિક ક્રિમતો આપવી

પ્રોગ્રામનો અમલ કરતી વખતે કોષ્ટક 15.1માં દર્શાવેલ કોષ્ટકના એરેને પ્રારંભિક ક્રિમતો આપવાનું કાર્ય કોડ-લિસ્ટિંગ 15.1માં આપેલ પ્રોગ્રામ દ્વારા કરી શકાય.

```
int sales[3][5], row, column;
for( row = 0; row < 3; row++) {
    for (column = 0; column < 5; column++){
        printf("Enter sales item [%d][%d]:", row, column);
        scanf("%d", &sales[row][column]);
    }
}
```

કોડ-લિસ્ટિંગ 15.1 : અમલ દરમિયાન ડેટા ઉમેરવાનો પ્રોગ્રામ

આ પ્રોગ્રામ ત્રણ આડી હરોળ અને પાંચ ઊભી હરોળ સાથેનું sales નામનું દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કરે છે. for લૂપ દ્વારા આપેલ તર્ક, sales નામના એરે માટે વિવિધ કુલ પંદર ક્રિમતો વાંચશે. sales એરેના અલગ-અલગ ઘટકોને કોડ-લિસ્ટિંગ 15.2 માં આપેલ પ્રોગ્રામ દ્વારા સ્ક્રીન પર દર્શાવી શકાય.

```
for( row = 0; row < 3; row++) {
    for (column = 0; column < 5; column++){
        printf("[%d][%d] : = %d \n", row, column, sales[row][column]);
    }
}
```

કોડ-લિસ્ટિંગ 15.2 : એરેની માહિતી દર્શાવવા માટેનો પ્રોગ્રામ

દ્વિપરિમાણીય એરેનું ઉદાહરણ

હવે આપણે પ્રોગ્રામના અમલ દરમિયાન એકપરિમાણીય અને દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કરવા અને તેની કિંમતોનો ઉપયોગ દર્શાવતો પ્રોગ્રામ સમજાએ. આ પ્રોગ્રામ વિદ્યાર્થીઓના ગુણને સાચવવા માટે દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કરશે. આ પ્રોગ્રામની સૂચનાઓ તેના પરિણામસહિત કોડ-લિસ્ટિંગ 15.3માં આપેલ છે.

```
/* Example 3 : Program to illustrate use of one અને two dimensional array. The program will maintain marks of two students in three quizzes using two dimensional arrays. The total marks of each student will be displayed.*/
```

```
#include<stdio.h>
```

```
#define STD 2 /*Number of rows for student data*/
```

```
#define QUIZ 3 /*Number of columns for quiz data*/
```

```
int main(){
```

```
    int marks[STD][QUIZ]; /* Marks array for 2 students અને 3 quizzes */
```

```
    int i, j, total_quiz[STD] = {0};
```

```
    for( i = 0; i < STD; i++) /* Loop for student row */
```

```
    {
```

```
        for( j = 0; j < QUIZ; j++) /* Loop for quiz column */
```

```
        {
```

```
            printf("Enter marks of student %d in quiz %d: ", i+1, j+1);
```

```
            scanf("%d", &marks[ i ][ j ]); /* Reading marks of quiz */
```

```
            total_quiz[ i ] = total_quiz[ i ] + marks[ i ][ j ];
```

```
        }
```

```
    }
```

```
    /* Logic to print student no., quiz marks અને total... */
```

```
    for ( i = 0 ; i < QUIZ; i++) /* Loop for printing quiz number... */
```

```
    {
```

```
        printf(" \tQuiz %d" , i+1);
```

```
    }
```

```
    printf(" Total \n");
```

```
    for( i = 0; i < STD; i++) /* Loop for student row */
```

```
    {
```

```
        printf("Student %d:", i+1);
```

```
        for( j = 0; j < QUIZ; j++) /* Loop for quiz column */
```

```
        {
```

```
            printf("%d \t", marks[ i ][ j ]);
```

```
        }
```

```
        printf("%d \n", total_quiz[ i ]);
```

```
    }
```

```
    return 0;
```

```
}
```

```
/* End of Program*/
```

/*~~~~~*/

Output:

Enter marks of student 1 in quiz 1 : 20
Enter marks of student 1 in quiz 2 : 30
Enter marks of student 1 in quiz 3 : 40
Enter marks of student 2 in quiz 1 : 50
Enter marks of student 2 in quiz 2 : 40
Enter marks of student 2 in quiz 3 : 20

	Quiz1	Quiz2	Quiz3	Total
Student 1 :	20	30	40	90
Student 2 :	50	40	20	110

/*~~~~~*/

કોડ-લિસ્ટિંગ 15.3 : ઉદાહરણ 15.3ની સૂચનાઓ અને પરિણામ

સમજૂતી

અહીં, આપણે આપણા પ્રોગ્રામમાં બે પ્રતીકાત્મક અચલ STD અને QUIZનો ઉપયોગ કરેલ છે. પ્રોગ્રામ દ્વારા કેટલા વિદ્યાર્થીઓની સંખ્યાને સંભાળવાની છે તે STD દ્વારા નિયંત્રિત થાય છે. એ જ રીતે, પ્રોગ્રામ દ્વારા સંભાળવામાં આવનાર પ્રશ્નોની સંખ્યા QUIZ દ્વારા નક્કી થશે. આપણે, વિદ્યાર્થીઓના પ્રશ્નોના ગુણ સાચવવા marks નામનો દ્વિપરિમાણીય એરે વ્યાખ્યાયિત કર્યો છે. કોઈ એક વિદ્યાર્થી દ્વારા મેળવાયેલ તમામ પ્રશ્નોના કુલ ગુણ સાચવવા માટે total_quiz નામનો એકપરિમાણીય એરે વ્યાખ્યાયિત કરવામાં આવ્યો છે. total_quizને વ્યાખ્યાયિત કરતી વખતે જ તેના બધા ઘટકોને પ્રારંભિક કિંમત તરીકે શૂન્ય આપવામાં આવેલ છે. આપણી જરૂરિયાત અનુસાર for લૂપના વિધાનોને પુનરાવર્તિત કરવા માટે i અને j નામના બે ગણતરી દર્શક (counter) ચલ વ્યાખ્યાયિત કરવામાં આવ્યા છે.

વિદ્યાર્થીઓના પ્રશ્નોના ગુણ વાંચવા એક for લૂપની અંદર વળી બીજી for લૂપ(નેસ્ટેડ લૂપ)નો ઉપયોગ કરવામાં આવ્યો છે. લૂપની અંદર આપણે એક વિદ્યાર્થી દ્વારા તમામ પ્રશ્નોના કુલ ગુણની ગણતરી કરી છે. એ પછીની for લૂપનો ઉપયોગ પરિણામને સ્ક્રીન પર યોગ્ય રીતે રજૂ કરવા માટે કરવામાં આવ્યો છે. આપેલ પ્રોગ્રામ માત્ર બે વિદ્યાર્થીઓના ત્રણ પ્રશ્નોના ગુણને સાચવવા માટે જ સક્ષમ છે. પરંતુ, પ્રોગ્રામમાં STD અને QUIZની કિંમતો બદલીને આપણે આપણા પ્રોગ્રામને વધુ ક્રિયાશીલ બનાવી શકીએ.

એરેની સીમાની ચકાસણી

એરે એ C ભાષાની એક સક્ષમતા છે, પરંતુ અહીં એ નોંધવું અગત્યનું છે કે, C ભાષામાં એરેની સીમાની ચકાસણીની કોઈ વ્યવસ્થા નથી. એરેની સીમાની ચકાસણી એટલે કે પ્રોગ્રામમાં વ્યાખ્યાયિત કરેલ એરેની સીમાની ચકાસણી કરવી. એરે વ્યાખ્યાયિત કરતું નીચે આપેલું વિધાન ચકાસો.

int number[5];

આ વિધાન પાંચ પૂર્ણાંક કિંમતો સાચવતા number નામના એરેને વ્યાખ્યાયિત કરશે. આપણે number[0], number[1] number[4]નો ઉપયોગ કરીને અલગ-અલગ એરે ઘટકોની કિંમતોનો ઉપયોગ કરી શકીશું. C ભાષા, એરેના []માં કોઈ પણ ક્રમસંખ્યા લખવા માટે પ્રોગ્રામરને સત્તા આપે છે. એનો મતલબ, આપણે number [-1] અને number [6] અથવા અન્ય કોઈ પણ અમાન્ય અનુક્રમનો ઉપયોગ કરી શકીએ. એરેના અનુક્રમની સીમા ચકાસવા માટેની કોઈ સુવિધા ન હોવાને કારણે આપણને પ્રોગ્રામમાં કોઈ ભૂલ દર્શાવશે નહીં. પરંતુ, કદાચ પ્રોગ્રામ આપણને કિંમતરૂપે કોઈ ખોટી કિંમત આપશે અથવા પ્રોગ્રામ અધવચ્ચેથી બંધ (crash) થઈ જશે.

સારાંશ

આ પ્રકરણમાં આપણે C ભાષામાં એકપરિમાણીય અને દ્વિપરિમાણીય એરેના ઉપયોગ સંબંધિત અગત્યની લાક્ષણિકતાઓ શીખ્યા. આપણે એ યાદ કરવું જોઈએ કે, એરેનો ડેટા પ્રકાર C ભાષાનો કોઈ પણ માન્ય ડેટા પ્રકાર હોઈ શકે છે. એરેને વ્યાખ્યાયિત કરવામાં સબસ્ક્રિપ્ટ તરીકે કોઈ પણ પૂર્ણાંક કે પૂર્ણાંક પદાવલિ જ હોવી જરૂરી છે. એરેનો સબસ્ક્રિપ્ટ ક્રમ (index) શૂન્યથી શરૂ થાય છે. કેરેક્ટર એરેનો અંત ખાસ ચિહ્ન (null character) '\0' થી જ આવે છે. એરેની સીમાની ચકાસણીની કોઈ વ્યવસ્થા C ભાષામાં નથી. પ્રોગ્રામમાં માન્ય સબસ્ક્રિપ્ટ ક્રમનો ઉપયોગ કરવાની જવાબદારી પ્રોગ્રામરની છે.

સ્વાધ્યાય

1. પ્રોગ્રામમાં એકપરિમાણીય એરેનો ઉપયોગ કરવાના ફાયદા જણાવો.
2. એકપરિમાણીય અને દ્વિપરિમાણીય એરે વચ્ચેનો તફાવત સ્પષ્ટ કરો.
3. કંપાઈલ સમયે એરેના ઘટકોને પ્રારંભિક કિંમત આપવી અને પ્રોગ્રામના અમલ દરમિયાન એરેના ઘટકોને પ્રારંભિક કિંમત આપવી એટલે શું?
4. નીચેના પ્રોગ્રામ કોડમાં ભૂલો શોધો, જો ભૂલ જણાય તો તેને સુધારો અને તેનું પરિણામ જણાવો :

(a) `#include <stdio.h>`

```
int main( ) {  
    int num[3][3] = {{1, 2}, {3, 4}};  
    printf("%d \n", number[1][1]);  
    return 0;  
}
```

(b) `#include<stdio.h>`

```
void main()  
{  
    char str[ ] = 'INDIA'  
    printf("%c %c %c %c  c", str[0], str[1], str[2], str[0], str[4]);  
}
```

5. ખાલી જગ્યા પૂરો :

- (a) દ્વિપરિમાણીય એરેને આડી હરોળ અને _____ હોય છે.
- (b) એરે ઘટક _____ સ્મૃતિસ્થાનો રોકે છે.
- (c) એરેની પાછળ ચોરસ કોંસમાં લખવામાં આવતા અનુક્રમ (Index)ને _____ તરીકે પણ ઓળખવામાં આવે છે.
- (d) સબસ્ક્રિપ્ટ પૂર્ણાંક હોવો જોઈએ અથવા _____ પદાવલિ હોઈ શકે.
- (e) C ભાષાના કેરેક્ટર એરેમાં શાબ્દિક માહિતીનો ખાસ ચિહ્ન _____ દ્વારા અંત લાવવામાં આવે છે.

6. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો :

- (a) એરેના ઘટકોને પ્રારંભિક કિંમતો માત્ર પ્રોગ્રામનો અમલ કરતી વખતે જ આપી શકાય છે.
- (b) C ભાષામાં એરેના સબસ્ક્રિપ્ટની કિંમત 1થી શરૂ થાય છે.
- (c) બહુપરિમાણીય એરેનું સાદું સ્વરૂપ એકપરિમાણીય એરે છે.
- (d) C ભાષા string ડેટા ટાઇપનું સમર્થન કરે છે.
- (e) દ્વિપરિમાણીય એરેને વ્યાખ્યાયિત કરતી વખતે પ્રથમ કૌંસ કુલ આડી હારની સંખ્યાનો નિર્દેશ કરે છે.
- (f) `int temperature[6] = {35, 32, 42};` એ એરેને પ્રારંભિક કિંમતો આપવાની માન્ય પદ્ધતિ છે.
- (g) `int num[] = {{1, 1}, {2, 2}};` એ પ્રારંભિક કિંમતો આપવાની માન્ય પદ્ધતિ છે.

7. નીચેના પ્રશ્નોના આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ આપો :

(1) C ભાષામાં જો કોઈ એરેના કુલ ઘટકની સંખ્યા કરતાં મોટી સંખ્યાના સબસ્ક્રિપ્ટ દ્વારા માહિતી મેળવવાનો પ્રયત્ન કરવામાં આવે તો શું થશે?

- (a) ઘટકને શૂન્ય બનાવી દેવાશે.
- (b) કંપાઈલર ભૂલ દર્શાવશે.
- (c) પ્રોગ્રામ અધવચ્ચેથી અટકી પડે અથવા ખોટી કિંમત આપશે.
- (d) એરેનું કદ આપમેળે વધારી આપવામાં આવશે.

(2) નીચેના પ્રોગ્રામનું પરિણામ શું આવશે?

```
int num[5] = {1, 2, 3, 4, 5};
```

```
int i, j;
```

```
i = num[1];
```

```
j = num[2];
```

```
printf("%d, %d, %d", i, j, num[0]);
```

- (a) 1, 2, 3 (b) 2, 3, 1 (c) 1, 2, 0 (d) 3, 4, 5

(3) નીચેના પ્રોગ્રામનું પરિણામ શું આવશે?

```
int num[5] = {1, 2, 3};
```

```
printf("%d, %d", num[0], num[3]);
```

- (a) 1, 2 (b) 1, 0 (c) 1, 3 (d) 2, 3

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો પાર પાડવા માટેનો C પ્રોગ્રામ લખો :

1. ઉપયોગકર્તા પાસેથી 5×5 નાં બે કોષ્ટક વાંચો. આ બે કોષ્ટકોનો સરવાળો કરી ત્રીજું કોષ્ટક બનાવી તેનું પરિણામ સ્ક્રીન પર દર્શાવો.
2. નીચે દર્શાવ્યા મુજબ પ્રોગ્રામમાં એરેનો ઉપયોગ કરી 20 વિદ્યાર્થીઓના ગુણ વાંચો :

Roll No.	Marks
1	60
2	70
...
20	75

- બધા પૈકી સૌથી વધુ ગુણ ધરાવતા વિદ્યાર્થીનો રોલ નંબર દર્શાવો.
 - 50 કરતાં વધુ ગુણ ધરાવતા કુલ વિદ્યાર્થીઓની સંખ્યા દર્શાવો.
 - 50 કરતાં ઓછા ગુણ ધરાવતા કુલ વિદ્યાર્થીઓની સંખ્યા દર્શાવો.
3. એકપરિમાણીય એરેમાં 10 પૂર્ણાંક સંખ્યા વાંચો. એરેના ઘટકોને અવળા ક્રમમાં દર્શાવો. ઉદાહરણ તરીકે, જો ઉપયોગકર્તાનું ઇનપુટ 10, 20, 30, 40 હોય તો પરિણામ 40, 30, 20, 10 એમ આવવું જોઈએ.
 4. એરેના N ઘટકોમાંથી મહત્તમ (maximum), લઘુત્તમ (minimum) અને સરેરાશ શોધો.
 5. ઉપયોગકર્તા પાસેથી 3×3 ના કોષ્ટકની પૂર્ણાંક સંખ્યા વાંચો. કોષ્ટકના દરેક ઘટકનો સરવાળો ગણીને દર્શાવો.





વિધેય

C ભાષાની અનેક ક્ષમતાઓ પૈકીની એક ક્ષમતા છે વિધેય (funcion) બનાવવા અને તેનો ઉપયોગ કરવો. વિધેય એ C ભાષાના કેટલાંક માન્ય વિધાનોનો સમૂહ છે, જે કોઈ ચોક્કસ કાર્ય પાર પાડે છે. વિધેય (funcion) ને પદ્ધતિ (method) સબરૂટિન (sub-routine) અથવા પ્રક્રિયા (procedure) તરીકે પણ ઓળખવામાં આવે છે. અત્યાર સુધી આપણે આપણા પ્રોગ્રામમાં C ભાષાના અસંખ્ય વિધેયોનો ઉપયોગ કર્યો છે. આપણે `main()`, `printf()` અને `scanf()` જેવા વિધેયોનો ઉપયોગ કરવાથી પરિચિત છીએ. તે C ભાષાના વિધેયોનું જ ઉદાહરણ છે. ચલાવી શકાય તેવા (executable) કોઈ પણ C પ્રોગ્રામમાં ઓછામાં ઓછું એક વિધેય `main()` તો હોય જ છે.

કોઈ પણ પ્રોગ્રામમાં વિધેયનો ઉપયોગ તે પ્રોગ્રામને મોડ્યુલર (modular) બનાવે છે. મોડ્યુલારિટી (Modularity) એટલે, કોઈ પણ જટિલ સમસ્યાને નાની નાની પેટા-સમસ્યાઓમાં વિભાજિત કરવી, જે સમજવામાં અને જાળવણીમાં સરળ બને. એકવાર કોઈ સમસ્યાને નાની નાની પેટા સમસ્યાઓમાં વિભાજિત કરી દેવાય એ પછી આપણે એ દરેક પેટા સમસ્યા માટે વિધેય લખી શકીએ. એ પછી, પ્રોગ્રામમાં બધા વિધેયોનો એક સાથે ઉપયોગ કરવાથી સમસ્યાનો ઉકેલ મળી જશે. આ પ્રકરણમાં આપણે C ભાષાના વિવિધ પ્રકારના વિધેયોની ચર્ચા કરીશું. પ્રોગ્રામમાં વિધેયને કેવી રીતે વ્યાખ્યાયિત કરવા અને કેવી રીતે ઉપયોગમાં લેવા તે પણ શીખીશું.

C ભાષામાં વિધેયોના બે પ્રકાર છે : (1) લાઈબ્રેરી વિધેય અથવા સિસ્ટમ ડિફાઈન્ડ વિધેય (2) યુઝર ડિફાઈન્ડ વિધેય

C ભાષાની સ્ટાન્ડર્ડ લાઈબ્રેરીમાં એવા ઘણા આંતરપ્રસ્થાપિત વિધેયો ઉપલબ્ધ છે, જેને આપણે આપણા પ્રોગ્રામમાં ઉપયોગમાં લઈ શકીએ તેને સિસ્ટમ ડિફાઈન્ડ ફંક્શન તરીકે પણ ઓળખવામાં આવે છે. ઉદાહરણ તરીકે `scanf()`, `printf()`, `sqrt()` અને `cos()` એ લાઈબ્રેરી વિધેયનાં ઉદાહરણ છે. સામાન્ય ઉપયોગકર્તા દ્વારા લાઈબ્રેરી વિધેયો લખવાની જરૂર નથી. આ વિધેયો ઉપયોગમાં લેવા C ભાષાની લાયબ્રેરીમાં કંપાઈલ (compile) કરેલા તૈયાર આપવામાં આવેલાં હોય છે. ઉદાહરણ તરીકે, `sqrt()` અને `cos()` એ ગણિત સંબંધી છે અને તેથી તેને `<math.h>` નામની હેડર (header) ફાઈલમાં મૂકવામાં આવેલ છે. કોઈ પણ લાઈબ્રેરી વિધેયનો ઉપયોગ કરવા માટે, આપણે તેની સંબંધિત હેડર ફાઈલને આપણા પ્રોગ્રામમાં સામેલ કરવી પડે. કેટલીક જાણીતી હેડર ફાઈલની યાદી અને તે ફાઈલમાંના વિધેયો આ પુસ્તકના પરિશિષ્ટ-IVમાં આપેલ છે.

તો ચાલો, ઉદાહરણ 16.1 મારફત એકાદ લાઈબ્રેરી વિધેયને સમજીએ. આ પ્રોગ્રામ આંતરપ્રસ્થાપિત લાઈબ્રેરી વિધેય `pow()`નો ઉપયોગ કરે છે, જે x નો ઘાતાંક y પરત આપે છે. આકૃતિ 16.1 કોડ-લિસ્ટિંગ અને ઉદાહરણ 16.1નું પરિણામ આપે છે. ઉદાહરણ 16.1માં આપણે `printf()` વિધેયનો ઉપયોગ કરેલ છે, જે `<stdio.h>` હેડર ફાઈલમાં વ્યાખ્યાયિત કરેલ છે. આપણે `pow()` વિધેયનો ઉપયોગ કરેલ છે, જે 8 નો ઘાતાંક 2 કિંમત પરિણામ તરીકે આપે છે. `pow()` વિધેય `<math.h>` ફાઈલમાં વ્યાખ્યાયિત કરેલ છે.

```

13-1.c - SciTE
File Edit Search View Tools Options Language Buffers Help

1 13-1.c
- /* Example 1: Program to illustrate use of pow() library
   funtion */
#include<stdio.h>
#include<math.h>
int main()
- {
    float answer;

    answer = pow(8,2);
    printf("The value of 8 raised to 2 is %f\n",answer);

    return 0;
}

>gcc -pedantic -Os -std=c99 13-1.c -o 13-1
>Exit code: 0
>./13-1
The value of 8 raised to 2 is 64.000000
>Exit code: 0

```

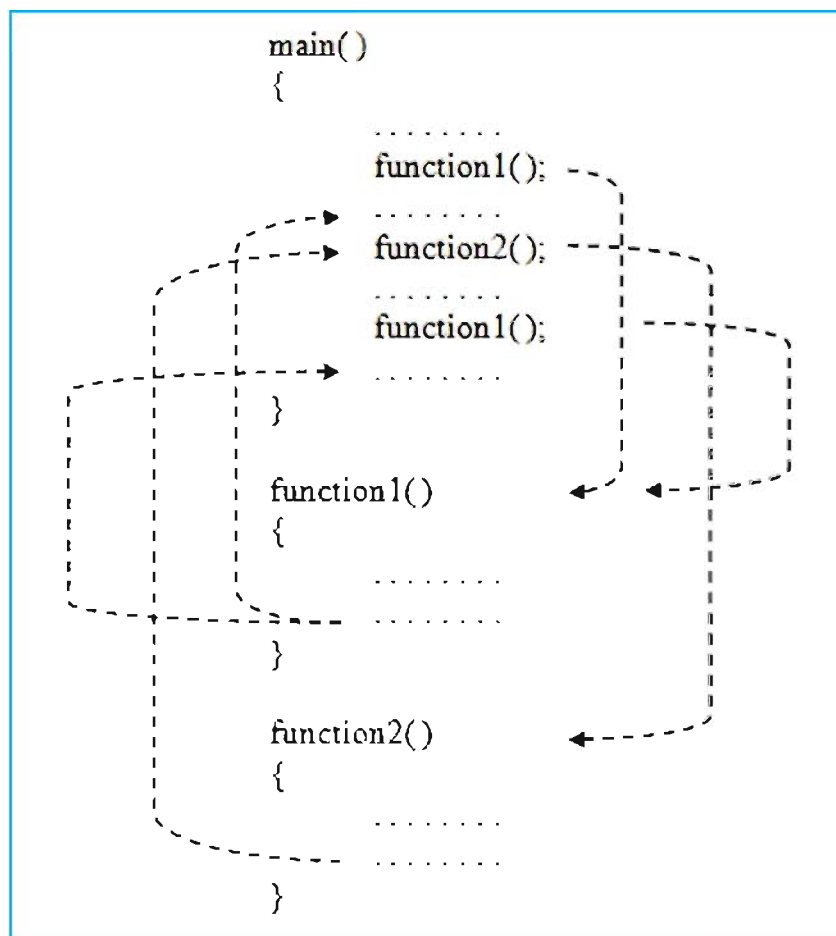
આકૃતિ 16.1 : ઉદાહરણ 16.1નું કોડ-લિસ્ટિંગ અને પરિણામ

ઉપયોગકર્તા દ્વારા તૈયાર કરવામાં આવેલા વિધેયોને યુઝર ડિફાઇન્ડ (user defined) વિધેય તરીકે ઓળખવામાં આવે છે. `main()` એ C ભાષામાં ખાસ પ્રકારનું યુઝર ડિફાઇન્ડ વિધેય છે. પ્રોગ્રામનો અમલ `main()` વિધેયમાંથી શરૂ થાય છે. જો આપણે આપણો પ્રોગ્રામ માત્ર `main()` વિધેયનો ઉપયોગ કરીને લખીએ, તો તે કદાચ મોટો બની જાય. આવા પ્રોગ્રામની ચકાસણી કરવી અને સુધારા-વધારા કરવા કદાચ થોડા મુશ્કેલ બની જાય. જો પ્રોગ્રામને નાના નાના સબપ્રોગ્રામ(વિધેય)માં વહેંચી દેવામાં આવે તો તેને સમજવામાં, ચકાસવામાં અને જરૂરી સુધારા-વધારા કરવાનું ઘણું સરળ બની જાય.

આપણા પ્રોગ્રામમાં વિધેયોનો ઉપયોગ કરવાના ફાયદા નીચે મુજબ છે :

- પ્રોગ્રામને વિધેયોમાં વિભાજિત કરવાથી પ્રોગ્રામ સમજવામાં અને સંભાળવામાં સરળ બને છે.
- એકવાર લખેલા વિધેયને પ્રોગ્રામ દ્વારા જરૂરિયાત અનુસાર અનેકવાર ઉપયોગમાં લઈ શકાય છે. આને કારણે પ્રોગ્રામ ટૂંકો બને છે અને એ રીતે મેમરીમાં બચત થાય છે.
- વિધેયને લીધે એક જ પ્રકારનું પ્રોગ્રામિંગ કાર્ય બેવડાતું અટકે છે.
- એક પ્રોગ્રામ માટે લખેલ વિધેય અન્ય કોઈ પણ પ્રોગ્રામ દ્વારા ઉપયોગમાં લઈ શકાય છે.
- જુદી જુદી કિંમતો આપીને એ જ વિધેયનો વારંવાર ઉપયોગ કરી શકાય છે.

તો ચાલો, હવે આપણા પ્રોગ્રામમાં જુદા જુદા વિધેયોનો કેવી રીતે ઉપયોગ કરવો તે સમજીએ. આકૃતિ 16.2 અનેક વિધેયોનો ઉપયોગ કરતાં પ્રોગ્રામના પ્રવાહનું નિયમન દર્શાવે છે.



આકૃતિ 16.2 : અનેક વિધેયવાળા પ્રોગ્રામના પ્રવાહનું નિયમન

આકૃતિ 16.2 એવું દર્શાવે છે કે, main() વિધેયમાં function1 બે વાર બોલાવવામાં આવે છે, અને function2 એક વાર બોલાવવામાં આવે છે. main()માં વિધાનોના અમલ દરમિયાન જ્યારે નિયંત્રણ function1() નામના વિધાન પર પહોંચે છે ત્યારે, નિયંત્રણ function1() માં પરિવર્તિત થાય છે. function1()નાં બધાં વિધાનોનો અમલ કર્યા પછી નિયંત્રણ ફરી main() વિધેયની અંદરના function1() વિધાન પછીના વિધાન પર પહોંચી જશે. એ જ રીતે, main() વિધેયમાંથી ફરી function2() અને એ પછી function1() ને બોલાવવામાં આવશે.

વિધેયને વ્યાખ્યાયિત કરવો

જો આપણે આપણા પ્રોગ્રામમાં ઉપયોગકર્તા દ્વારા વ્યાખ્યાયિત વિધેયનો ઉપયોગ કરવા ઇચ્છતા હોઈએ તો આપણે આપણા કંપાઈલર (compiler)ને તેની જાણ કરવી પડે. આ માટે આપણે વિધેયને વ્યાખ્યાયિત કરતું વિધાન લખીને કંપાઈલરને જાણ કરી શકીએ. અગાઉથી વ્યાખ્યાયિત કર્યા વગર વિધેયનો ઉપયોગ કરી શકાય નહીં. C ભાષામાં નીચે મુજબની વાક્યરચનાનો ઉપયોગ કરીને કોઈ પણ વિધેયને વ્યાખ્યાયિત કરી શકાય છે.

```
return_data_type  function name (arguments)

{

    function statements;

}
```

અહીં return_data_type એ વિધેય દ્વારા પરત કરવામાં આવતી માહિતીનો પ્રકાર નક્કી કરે છે. જો વિધેય દ્વારા પરત આપવામાં આવતી કિંમત પૂર્ણાંક હોય તો return_data_type તરીકે int હોય. જો વિધેય કોઈ પણ કિંમત પરત ન કરે તેમ હોય તો return_data_type તરીકે void હોઈ શકે. functionname યુઝર ડિફાઈન્ડ વિધેયનું નામ નક્કી કરે છે. વિધેયને નામ આપવાના નિયમો, ચલનું નામ આપવાના નિયમો જેવા જ છે. વિધેયનાં નામ બને ત્યાં સુધી અર્થસભર આપવાં જોઈએ. ઉદાહરણ તરીકે સરવાળા કરવા માટે add(), સરેરાશ ગણવા માટે avg() વગેરે. arguments વિધેયને આપવામાં આવતી ડેટા પ્રકાર સહિતની ઇનપુટ કિંમતો દર્શાવે છે. એક કરતા વધુ વિધેયને વ્યાખ્યાયિત કરતી વખતે એક કરતા વધુ આર્ગ્યુમેન્ટ (સંલગ્ન મૂલ્ય) માટે બધાં આર્ગ્યુમેન્ટ અલ્પવિરામ દ્વારા અલગ પાડવામાં આવે છે. બે છગડિયા કૌંસની વચ્ચેનાં વિધાનોને ‘ફંક્શન બોડી’ (function body) કહે છે. પ્રોગ્રામની જરૂરિયાતને આધારે ફંક્શન બોડીમાં સ્થાનિક ચલની ઘોષણા અને return વિધાન હોઈ શકે. વિધેયમાં ચલની ઘોષણા અને return વિધાન આપણે આ પ્રકરણમાં હવે પછી ચર્ચીશું.

અહીં એ નોંધો કે, લાઈબ્રેરી વિધેયને વ્યાખ્યાયિત કરવાની કોઈ જરૂર હોતી નથી, કારણ કે તે અગાઉ વ્યાખ્યાયિત થયેલ જ હોય છે. કોઈ પ્રોગ્રામ દ્વારા ઉપયોગમાં લેતા પહેલાં, બધા યુઝર ડિફાઈન્ડ વિધેય, વિધેય પ્રતિકૃતિ (function prototype) તરીકે વ્યાખ્યાયિત થવા જરૂરી છે. વિધેય પ્રોટોટાઈપ એટલે, main () વિધેયમાં ઉપયોગ પૂર્વ વિધેયને વ્યાખ્યાયિત કરવો.

વિધેયને બોલાવવા

અગાઉ જણાવ્યા પ્રમાણે, દરેક C પ્રોગ્રામ main () વિધેય સાથે ચાલુ થાય છે. main () વિધેયમાંથી આપણે બીજા યુઝર ડિફાઈન્ડ વિધેય અથવા લાઈબ્રેરી ફંક્શન બોલાવીએ છીએ. કોઈ પણ વિધેયને ઉપયોગ માટે બોલાવવા જરૂરિયાત મુજબના પેરામીટર સાથે તે વિધેયના નામનો ઉપયોગ કરો. તો ચાલો, ઉદાહરણ 16.2 જુઓ, જે દર્શાવે છે કે main()માંથી કોઈ વિધેયને કેવી રીતે બોલાવવા. આકૃતિ 16.3, ઉદાહરણ 16.2નું કોડ-લિસ્ટિંગ આપે છે, તેનું પરિણામ આકૃતિ 16.4માં દર્શાવાય છે.

```
13-2.c - SciTE
File Edit Search View Tools Options Language Buffers Help

13-2.c
/* Example 2: Program to illustrate use of function which returns no value.*/
#include<stdio.h>

void print_line( );           // Function Prototype Declaration

void main( )
{
    print_line( );           // Function call
    printf("...Wel Come to the World of C Function...\n");
    print_line( );           // Function call
}
// End of main() function

void print_line( )
{
    printf(" = = = = = \n");
}
// End of print_line() function
```

આકૃતિ 16.3 : ઉદાહરણ 16.2નું કોડ-લિસ્ટિંગ

```
gcc 13-2.c
>gcc 13-2.c
>Exit code: 0
./a.out
>./a.out
= = = = =
...Wel Come to the World of C Function...
= = = = =
>Exit code: 45
```

આકૃતિ 16.4 : ઉદાહરણ 16.2નું પરિણામ

સમજૂતી

પ્રોગ્રામમાં `print_line()` નામના એક યુઝર ડિફાઈન્ડ વિધેયનો ઉપયોગ કરવામાં આવ્યો છે. `main()` વિધેયમાં લખેલું `print_line()` વિધાન આ વિધેયને બોલાવે છે. પ્રોગ્રામનું નિયંત્રણ `main()` વિધેય પાસેથી `print_line()` વિધેય પાસે જાય છે, જ્યાં તેના દ્વારા વિવિધ શૈલીની લીટીઓ છાપીને પ્રોગ્રામનું નિયંત્રણ `main()` વિધેયમાં પાછું આવી જાય છે. `main()` વિધેયમાં એ પછીનું `printf` વિધાન એક સંદેશ છાપે છે. `main()` વિધેયનું એના પછીનું વિધાન ફરીથી `print_line()` વિધેયને બોલાવે છે. આમ આ ઉદાહરણમાં `print_line()` વિધેયને બે વાર બોલાવવામાં આવે છે. આ વિધેય `main()` વિધેયને કોઈ કિંમત પરત આપતું ન હોવાથી તેનો પરત પ્રકાર (return type) `void` છે.

Return વિધાન

યુઝર ડિફાઈન્ડ વિધેય તેને બોલાવનાર વિધેયને કદાચ કોઈ કિંમત પરત કરે કે ન પણ કરે. બોલાવનાર વિધેયને કોઈ પણ કિંમત પરત કરવા માટે ફંક્શન બોડીની અંદર આપણે `return` વિધાનનો ઉપયોગ કરી શકીએ. `return` વિધાનનું સ્વરૂપ નીચે મુજબ છે :

`return;` અથવા `return (expression);`

return વિધાનનું પ્રથમ સ્વરૂપ કોઈ પણ કિંમત પરત આપશે નહીં. તે માત્ર તેને બોલાવનાર પ્રોગ્રામને પ્રોગ્રામનું નિયંત્રણ પાછું સોંપે છે. જ્યારે કોઈ વિધેયની 'રિટર્ન ટાઇપ' (return type) void હોય તો તે વિધેયની ફંક્શન બોડીમાં return વિધાન લખવાની કોઈ જરૂર રહેતી નથી.

કોઈ વિધેય તેને બોલાવનાર વિધેયને કેવી રીતે કિંમત પરત કરે છે તે સમજવા આકૃતિ 16.5 પર એક નજર નાંખો, જે ઉદાહરણ 16.3નું કોડ-લિસ્ટિંગ દર્શાવે છે.

સમજૂતી

આકૃતિ 16.5માં આપણે add() નામના ચુઝર ડિફાઇન્ડ વિધેયનો ઉપયોગ કર્યો છે, main() વિધેયમાં આપણે total નામના એક ચલને વ્યાખ્યાયિત કર્યો છે. આપણે જ્યારે main()માંથી add() વિધેયને બોલાવીએ છીએ ત્યારે પ્રોગ્રામનું નિયંત્રણ add() વિધેય પાસે જાય છે. add વિધેયમાં num1 અને num2 નામના બે સ્થાનિક ચલ ઘોષિત કરવામાં આવેલ છે. add વિધેયનું એ પછીનું વિધાન સંદેશ દર્શાવે છે અને અનુક્રમે printf અને scanf વિધાનનો ઉપયોગ કરીને બે સંખ્યા વાંચે છે. છેલ્લું વિધાન બે સંખ્યાઓનો સરવાળો પરત આપે છે, જે main() વિધેયના total નામના ચલમાં સાચવવામાં આવે છે. એ પછી totalને સ્ક્રીન પર દર્શાવવામાં આવે છે.

```
13-3.c - SciTE
File Edit Search View Tools Options Language Buffers Help

13-3.c

/* Example 3: Program to illustrate use of function which returns a value.*/

#include<stdio.h>
int add( );           // Function Prototype Declaration

void main( )
{
    int total;
    total = add( );    // Function call
    printf("The addition of two numbers: %d \n", total);
}
// End of main() function...

int add( )
{
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    return(num1 + num2);
}
// End of add() function...
```

આકૃતિ 16.5 : ઉદાહરણ 16.3નું કોડ-લિસ્ટિંગ

```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 13-3.c
kpp@ubuntu:~$ ./a.out
Enter two numbers: 20 10
The addition of two numbers: 30
kpp@ubuntu:~$ 

```

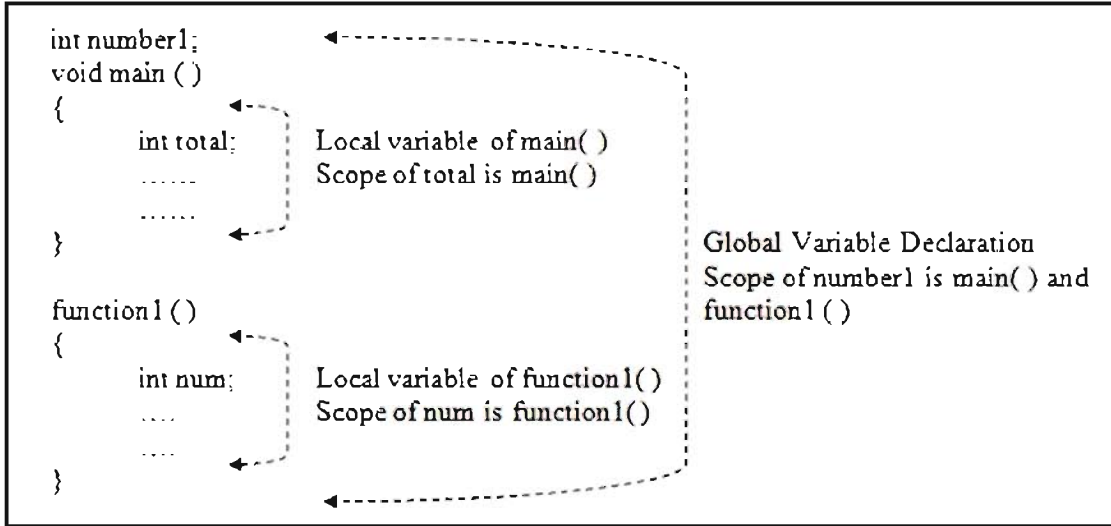
આકૃતિ 16.6 : ઉદાહરણ 16.3નું પરિણામ

ચલનો અવકાશ (Scope of Variables)

ચલના અવકાશ(scope)નો અર્થ છે પ્રોગ્રામના ક્યા ભાગમાં ચલનો ઉપયોગ કરી શકાય છે. C ભાષામાં ચલના અવકાશના બે પ્રકાર છે:

- સાર્વત્રિક ચલ (Global Variable)
- સ્થાનિક ચલ (Local Variable)

આકૃતિ 16.7, C ભાષાના પ્રોગ્રામમાં ચલના અવકાશને સમજાવે છે.



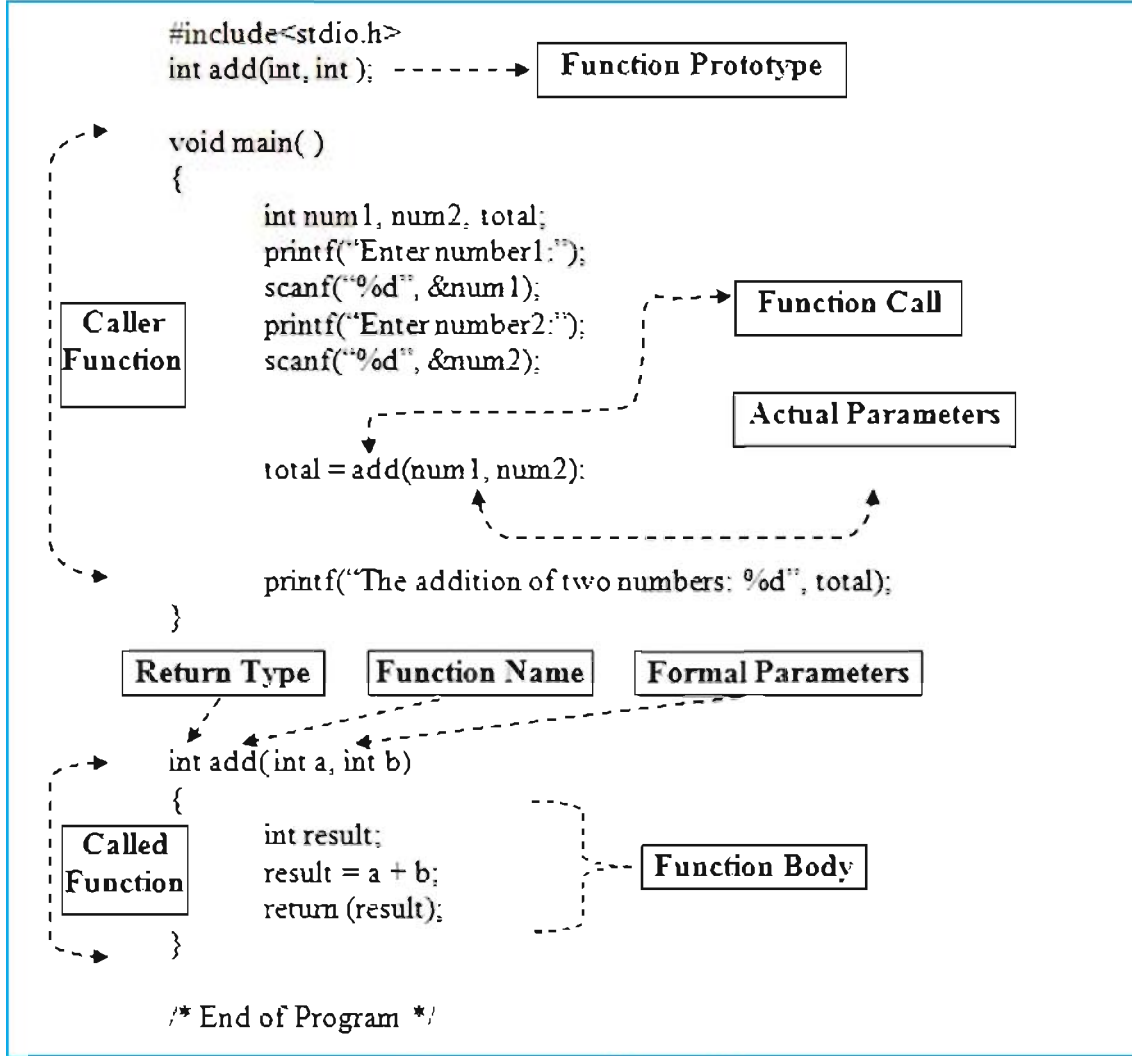
આકૃતિ 16.7 : ચલનો અવકાશ

અહીં number1 નામનો ચલ એ સાર્વત્રિક ચલ ગણાશે, કારણ કે પ્રોગ્રામમાં તેને main () વિધેયની ઉપર વ્યાખ્યાયિત કરવામાં આવ્યો છે. આ સાર્વત્રિક ચલ number1 ને main () ફંક્શનમાં અને function1 () માં એમ બંને જગ્યાએ ઉપયોગમાં લઈ શકાશે. total નામનો સ્થાનિક ચલ માત્ર main () વિધેયમાં જ ઉપયોગમાં લઈ શકાશે, જ્યારે function1 () વિધેયમાં તેને ઉપયોગમાં લઈ શકાશે નહીં. એ જ રીતે, સ્થાનિક ચલ num માત્ર function1 () નામના વિધેય દ્વારા ઉપયોગમાં લઈ શકાશે પણ main () વિધેયમાં ઉપયોગમાં લઈ શકાશે નહીં.

C ભાષાના વિધેયોની વિસ્તૃત લાક્ષણિકતાઓની ચર્ચા કરીએ તે પહેલાં ચાલો આપણે આકૃતિ 16.8 જોઈએ, જે વિધેયના વિવિધ વિભાગ દર્શાવે છે.

આકૃતિ 16.8માં દર્શાવ્યા પ્રમાણે આપણે int add(int, int); એ પ્રમાણે વિધેયની પ્રતિકૃતિ વ્યાખ્યાયિત કરેલ છે. આ દર્શાવે છે કે અહીં પૂર્ણાંક (integer) પ્રકારના બે આર્ગ્યુમેન્ટ છે. વિધેયની પ્રતિકૃતિ વિશે વિસ્તૃત ચર્ચા પછી આ પ્રકરણમાં કરવામાં આવી છે. void main () { } એ બોલાવનાર વિધેય (caller function) છે જે બે ખરા પેરામીટર num1 અને num2 સાથે add () વિધેયને બોલાવે છે. અહીં int add () નામનું યુઝર ડિફાઇન્ડ વિધેય એવું દર્શાવે છે કે, તે તેને બોલાવનાર વિધેય void main () ને પૂર્ણાંક પરત કરશે. add વિધેયમાં બે

ઔપચારિક પૂર્ણાંક પેરામીટર "a" અને "b" વ્યાખ્યાયિત કરવામાં આવ્યા છે. આ ઔપચારિક ચલ, main() માંથી વિધેયને બોલાવતી વખતે મોકલવામાં આવતી કિંમતોનો સ્વીકાર કરે છે. result નામનો પૂર્ણાંક ચલ, બોલાવનાર વિધેયને બે સંખ્યાનો સરવાળો પરત આપે છે, જે main()ના total નામના ચલમાં સાચવવામાં આવે છે.



આકૃતિ 16.8 : C વિધેયના વિભાગ

વિધેયની પ્રતિકૃતિ

કોઈ પણ પ્રોગ્રામમાં જ્યારે કોઈ વિધેયને `main()` પછી વ્યાખ્યાયિત કરવામાં આવે ત્યારે વિધેયની પ્રતિકૃતિ (function prototype)ની આવશ્યકતા ઊભી થાય છે. આપણે જ્યારે આપણા પ્રોગ્રામમાં કોઈ પણ વિધેયને બોલાવીએ છીએ, ત્યારે કંપાઈલર આવા સંબંધિત વિધેયની વ્યાખ્યાને શોધે છે. આવું એ ચક્રસવા માટે થાય છે કે વિધેયને બોલાવવાની ક્રિયા ખરી છે કે નહીં. જો બોલાવવામાં આવેલ વિધેય આર્ગ્યુમેન્ટની સંખ્યા અને તેના પ્રકારની બાબતે વિધેયના મથાળાની લીટી (header of the function) સાથે બંધબેસતી હશે, તો તે વિધેય માન્ય ગણાશે અન્યથા કંપાઈલર દ્વારા ભૂલસંદેશ દર્શાવાશે. ઉદાહરણ તરીકે,

`int add(int, int);` જે `int add(int a, int b)` વિધેયની પ્રતિકૃતિ (prototype) છે.

અહીં પ્રતિકૃતિ એવું દર્શાવે છે કે, `add()` એ બે પૂર્ણાંક આર્ગ્યુમેન્ટ ધરાવતું એક વિધેય છે, અને તે પૂર્ણાંક કિંમત પરત આપે છે. વિધેયની પ્રતિકૃતિ દરેક ઉપયોગકર્તા નિર્મિત વિધેય માટે પ્રોગ્રામની શરૂઆતમાં જ લખવામાં આવે છે.

વિધેયનું સંલગ્ન મૂલ્ય અથવા પેરામીટર

વિધેયના સંલગ્ન મૂલ્ય અથવા પેરામીટરનો ઉપયોગ કરવા પાછળનું મૂળ પ્રયોજન બોલાવવામાં આવેલ અને બોલાવનાર વિધેય વચ્ચે ડેટાનું આદાન-પ્રદાન કરવાનું છે. જ્યારે બોલાવનાર વિધેય (calling function) બોલાવવામાં આવેલ વિધેય (called function) ને ડેટા મોકલે છે ત્યારે તેને વિધેયનું સંલગ્ન મૂલ્ય (arguments) અથવા પેરામીટરની તબદીલી (parameter passing) કહે છે.

વિધેયને વ્યાખ્યાયિત કરતી વખતે ઉપયોગમાં લેવાતા સંલગ્ન મૂલ્ય(arguments)ને “ઔપચારિક સંલગ્ન મૂલ્ય” (formal arguments) તરીકે ઓળખવામાં આવે છે. આવા વિધેયને બોલાવવાની જરૂર પડે ત્યારે બોલાવતી વખતે તેની સાથે એટલી જ સંખ્યામાં સંલગ્ન મૂલ્યોને તબદીલ કરવાની જરૂર પડે છે. આ તબદીલ કરવામાં આવતા સંલગ્ન મૂલ્યોને વાસ્તવિક સંલગ્ન મૂલ્યો (actual arguments) તરીકે ઓળખવામાં આવે છે. “ઔપચારિક સંલગ્ન મૂલ્ય” અને “વાસ્તવિક સંલગ્ન મૂલ્ય”ની સંખ્યા, પ્રકાર અને ક્રમ બાબતે એકબીજા સાથે મળતા હોવા જોઈએ. “વાસ્તવિક સંલગ્ન મૂલ્યો” જો “ઔપચારિક સંલગ્ન મૂલ્યો” કરતાં વધુ હશે તો વધારાના વાસ્તવિક સંલગ્ન મૂલ્યને છોડી દેવામાં આવશે. તેનાથી ઊલટું “વાસ્તવિક સંલગ્ન મૂલ્યો” જો “ઔપચારિક મૂલ્યો” કરતાં ઓછાં હશે તો બંધબેસતાં નહીં હોય તેવા “ઔપચારિક-સંલગ્ન મૂલ્યોને ખોટી કિંમત વડે પ્રારંભિક કિંમત અપાશે. એ જ રીતે, કોઈ પણ સંલગ્ન મૂલ્યોના ડેટા પ્રકારમાં કોઈ પણ જાતની વિસંગતતા ખોટી પ્રારંભિક કિંમત આપવા માટે કારણભૂત નીવડશે.

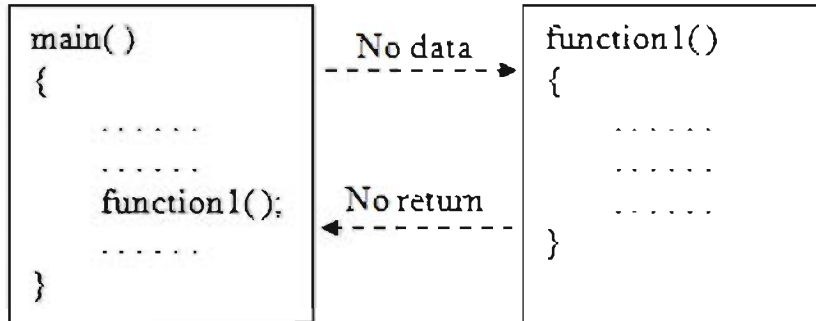
વિધેયના પ્રકાર

વિધેયમાં સંલગ્ન મૂલ્યોની ઉપસ્થિતિ અને પરત મળતી કિંમતને આધારે વિધેયોના નીચે પ્રમાણે ત્રણ પ્રકાર પાડી શકાય :

- કોઈ પણ સંલગ્ન મૂલ્ય કે પરત કિંમત ન આપતા વિધેય
- સંલગ્ન મૂલ્ય ધરાવતા પરંતુ પરત કિંમત વિનાના વિધેય
- સંલગ્ન મૂલ્ય અને પરત કિંમત ધરાવતા વિધેય

કોઈ પણ સંલગ્ન મૂલ્ય કે કોઈ પરત કિંમત ન આપતા વિધેય

જ્યારે કોઈ વિધેય કોઈ સંલગ્ન મૂલ્ય ન ધરાવતા હોય અને કોઈ પણ કિંમત પરત ન આપતા હોય તો તે આ પ્રકારના વિધેયમાં આવે છે. બોલાવવામાં આવેલ વિધેય(called function)ને બોલાવનાર વિધેય (calling function) તરફથી કોઈ પણ ડેટા કિંમતો પ્રાપ્ત થતી નથી. એ જ રીતે, કોઈ પરત કિંમત ન હોવાને લીધે બોલાવનાર વિધેય (calling function) ને પણ બોલાવવામાં આવેલ વિધેય (called function) તરફથી કોઈ પણ ડેટા પરત મળતો નથી. ટૂંકમાં, બોલાવનાર વિધેય અને બોલાવવામાં આવેલ વિધેય વચ્ચે કોઈ પણ જાતના ડેટાની આપ-લે થતી નથી. આ બાબત આકૃતિ 16.9માં દર્શાવેલ છે. આકૃતિમાં બે પ્રતીકાત્મક વિધેયની વચ્ચે દેખાતી ટપકાવાળી રેખા કોઈ ડેટાની તબદીલી દર્શાવતા નથી પરંતુ માત્ર પ્રોગ્રામના નિયંત્રણની તબદીલી દર્શાવે છે.

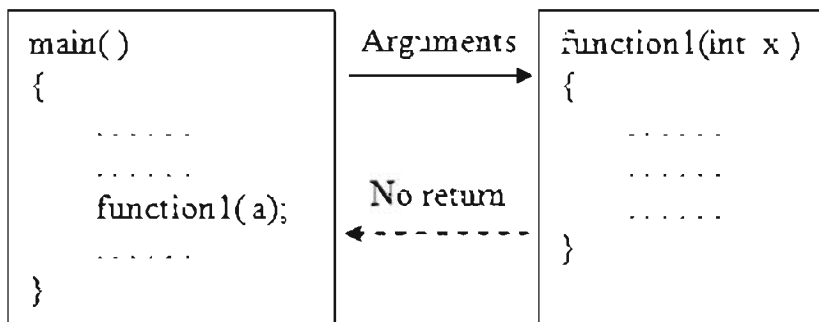


આકૃતિ 16.9 : કોઈ પણ ડેટાસંચાર વિનાનું વિધેય

આ પ્રકરણમાં અગાઉ સમજાવેલ ઉદાહરણ 16.2 એ આ પ્રકારના વિધેયનું ઉદાહરણ છે.

સંલગ્ન મૂલ્ય સહિતનું પરંતુ પરતમૂલ્ય વિનાનું વિધેય

બોલાવનાર વિધેય અને બોલાવવામાં આવેલ વિધેય વચ્ચે જરૂરી સંલગ્ન મૂલ્ય સાથે પરંતુ કોઈ પણ પરતમૂલ્ય વગરનું પ્રત્યાયન (communication) આકૃતિ 16.10માં દર્શાવેલ છે.



આકૃતિ 16.10 : એકતરફી ડેટાસંચાર ધરાવતું વિધેય

હવે, આપણે ઉદાહરણ 16.4 સમજીએ, જે ઉપયોગકર્તા પાસેથી એક સંખ્યા મેળવશે અને વિધેયનો ઉપયોગ કરી તેનું ઘનફળ (cube) શોધશે. આ માટે આપણે main() (બોલાવનાર) વિધેયમાંથી સંલગ્ન મૂલ્ય તરીકે એક સંખ્યાને cube() (બોલાવવામાં આવેલ) વિધેયને મોકલીશું. બોલાવવામાં આવેલ (called) વિધેય ઘનફળ શોધશે અને તેને સ્ક્રીન પર દર્શાવશે. આકૃતિ 16.11, ઉદાહરણ 16.4નું કોડ-લિસ્ટિંગ આપે છે, જ્યારે આકૃતિ 16.12 પરિણામ દર્શાવે છે.

```

13-4.c - SciTE
File Edit Search View Tools Options Language Buffers Help

13-4.c
/* Example 4: Program to illustrate use of function with arguments and no return value.*/
#include<stdio.h>

void cube ( int );           // Function Prototype Declaration

void main( )
{
    int number;
    printf("Enter your number: ", number);
    scanf("%d", &number);
    cube( number );         // Function call
}
//End of main () function

void cube(int num1)         // Function to find and display cube of a given number.
{
    printf("The cube of entered number is : %d \n", num1*num1*num1);
}

// End of cube() function

```

આકૃતિ 16.11 : ઉદાહરણ 16.4નું કોડ-લિસ્ટિંગ

```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 13-4.c
kpp@ubuntu:~$ ./a.out
Enter your number: 10
The cube of entered number is : 1000
kpp@ubuntu:~$ 

```

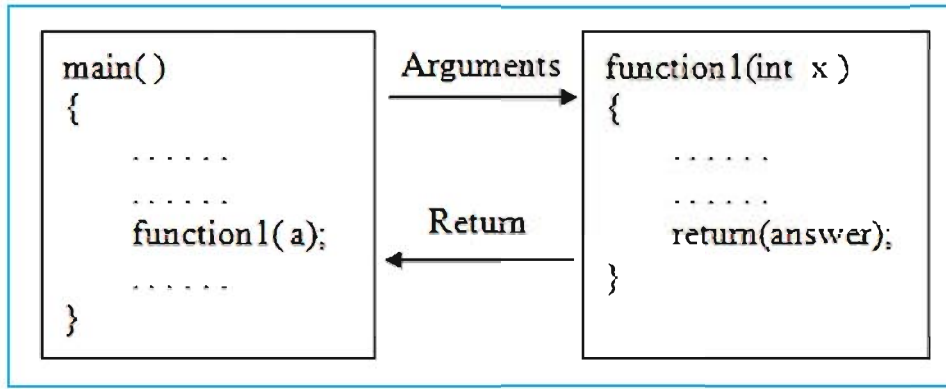
આકૃતિ 16.12 : ઉદાહરણ 16.4નું પરિણામ

સમજૂતી

અહીં main() વિધેય ઉપયોગકર્તા પાસેથી એક સંખ્યા સ્વીકારે છે, જે number() નામના ચલમાં સાચવવામાં આવે છે. આ number નામનો ચલ cube() નામના વિધેયને ખરેખરા સંલગ્ન મૂલ્ય તરીકે પહોંચાડવામાં આવે છે. cube() વિધેયમાં num1 નામનું ઔપચારિક સંલગ્ન મૂલ્ય number નામના ચલની કિંમત મેળવે છે. એ પછી, cube() વિધેય ગણતરી કરે છે અને printf વિધાનનો ઉપયોગ કરી આપેલ સંખ્યાનું ઘનફળ દર્શાવે છે.

સંલગ્નમૂલ્ય અને પરતમૂલ્ય સહિતના વિધેય

તો ચાલો બોલાવનાર વિધેય અને બોલાવવામાં આવેલ વિધેય વચ્ચે દ્વિમાર્ગીય સંચાર સમજીએ. સંલગ્ન મૂલ્ય અને પરત મૂલ્ય ધરાવતું વિધેય આકૃતિ 16.13માં દર્શાવેલ છે.



આકૃતિ 16.13 : દ્વિતરફી ડેટાસંચાર સાથેનું વિધેય

આપણે ઉદાહરણ 16.5 સમજવા પ્રયત્ન કરીએ, જે ઉપયોગકર્તા પાસેથી એક સંખ્યા મેળવશે અને `sqr()` વિધેયનો ઉપયોગ કરી તેનો વર્ગ (square) શોધશે. આપણે `main()` નામના બોલાવનાર વિધેયમાંથી બોલાવવામાં આવનાર વિધેય `sqr()` ને સંલગ્ન મૂલ્ય તરીકે એક સંખ્યા મોકલીશું. બોલાવેલ વિધેય સંખ્યાનો વર્ગ શોધીને તેને બોલાવનાર વિધેયને પરત મોકલે છે. આકૃતિ 16.14 ઉદાહરણ 16.5નું કોડ-લિસ્ટિંગ આપે છે, જ્યારે આકૃતિ 16.15 તેનું પરિણામ દર્શાવે છે.

```

13-5.c - SciTE
File Edit Search View Tools Options Language Buffers Help
13-5.c
/* Example 5: Program to illustrate use of function with arguments and return value.*/
#include<stdio.h>

int sqr ( int );           // Function Prototype Declaration

void main( )
{
    int number, result;
    printf("Enter your number: ", number);
    scanf("%d", &number);
    result = sqr( number );           // Function call with argument
    printf("The square of %d is %d. \n", number, result);
}
//End of main() function

int sqr(int num1) // Function to find square of a number and return the answer
{
    int answer;
    answer = num1 * num1;
    return (answer);
}
//End of sqr() function
  
```

આકૃતિ 16.14 : ઉદાહરણ 16.5નું કોડ-લિસ્ટિંગ

```

kpp@ubuntu: ~
File Edit View Terminal Help
kpp@ubuntu:~$ gcc 13-5.c
kpp@ubuntu:~$ ./a.out
Enter your number: 10
The square of 10 is 100.
kpp@ubuntu:~$ 
  
```

આકૃતિ 16.15 : ઉદાહરણ 16.5નું પરિણામ

સમજૂતી

main() વિધેય ઉપયોગકર્તા પાસેથી એક સંખ્યા મેળવે છે, જેને number નામના ચલમાં સાચવવામાં આવે છે. એ પછી, sqr() નામના વિધેયને number નામનો ચલ સંલગ્ન મૂલ્ય તરીકે મોકલવામાં આવે છે. sqr() વિધેયમાં ઔપચારિક સંલગ્ન મૂલ્ય num1, number નામના ચલની કિંમત મેળવે છે. એ પછી sqr() વિધેય વર્ગફળ ગણે છે અને તેને answer નામના ચલમાં સાચવે છે. sqr() વિધેયનું છેલ્લું વિધાન answer નામના ચલમાં સાચવેલી કિંમત બોલાવનાર વિધેયને પરત આપે છે. બોલાવનાર વિધેય પરત મળતી કિંમતને સ્વીકારે છે અને તેને result નામના ચલમાં સાચવે છે. main() વિધેયનું છેલ્લું વિધાન સ્ક્રીન પર આપેલ સંખ્યાનું વર્ગફળ દર્શાવે છે.

સારાંશ

આ પ્રકરણમાં આપણે C ભાષાના વિધેય સાથે સંબંધિત મૂળભૂત ખ્યાલોની ચર્ચા કરી. આપણે એ જોયું કે, પ્રોગ્રામમાં વિધેયનો ઉપયોગ તેને મોડ્યુલર (modular) બનાવે છે. લાઈબ્રેરી વિધેય અને ઉપયોગકર્તા નિર્મિત વિધેયોને આપણા પ્રોગ્રામમાં કેવી રીતે ઉપયોગમાં લેવા તે પણ આપણે શીખ્યા. એ જ રીતે આપણે એ પણ શીખ્યા કે, આપણા પ્રોગ્રામમાં સ્થાનિક ચલ અને સાર્વત્રિક ચલોનો ઉપયોગ કેવી રીતે કરવો ? આપણે એ જોયું કે કોઈ એક વિધેય કેવી રીતે, બોલાવનાર વિધેયને એક કિંમત પરત આપી શકે. વિધેયમાં સંલગ્ન મૂલ્ય કે પેરામીટરના ઉપયોગ પાછળના મૂળભૂત ઉદ્દેશની ચર્ચા કરી.

સ્વાધ્યાય

1. વિધેય એટલે શું? પ્રોગ્રામમાં વિધેયનો ઉપયોગ કરવાના ફાયદા જણાવો.
2. પ્રોગ્રામમાં લાઈબ્રેરી વિધેયનો ઉપયોગ કરવાના ફાયદા જણાવો.
3. ચલનો અવકાશ (scope of a variable) એટલે શું?
4. કોઈ પણ વિધેયને ડેટા / માહિતી કેવી રીતે પહોંચાડી શકાય?
5. વિધેયની પ્રતિકૃતિ (function prototype) એટલે શું?
6. નીચેના વચ્ચે તફાવત સ્પષ્ટ કરો :
 - (a) સ્થાનિક અને સાર્વત્રિક ચલ
 - (b) ઔપચારિક અને વાસ્તવિક પેરામીટર
 - (c) લાઈબ્રેરી વિધેય અને ઉપયોગકર્તા નિર્મિત વિધેય (User defined function)
7. નીચેના પ્રોગ્રામમાં જો કોઈ ભૂલ હોય તો તે શોધી કાઢી, તેને ફરીથી લખી પ્રોગ્રામનું પરિણામ શું આવશે તે જણાવો :
 - (a)

```
#include<stdio.h>
void draw_line( );
void main( )
{
    draw_line;
    printf("...Wel Come to the C Function...\n");
    draw_line;
}
void print_line( )
{
    printf("= = = = = \n");
}
```

(b)

```
#include<stdio.h>
int sqr ( int );
void main( )
{
    int number = 5, result;
    result = sqr( );
    printf("The square of 5 is %d.", result);
}
int sqr(int num1)
{
    int answer;
    answer = num1 * num1;
    print (answer);
}
```

8. ખાલી જગ્યા પૂરો :

- (a) વિધેયમાંથી _____ વિધાન માહિતીને પરત આપે છે.
- (b) જો કોઈ વિધેય કોઈ પણ કિંમત પરત ન કરતું હોય તો તે વિધેયની “રિટર્ન ડેટા ટાઈપ” _____ હોય.
- (c) `sqr()` નામનું લાઈબ્રેરી વિધેય _____ હેડર ફાઈલમાં વ્યાખ્યાયિત કરવામાં આવેલું છે.
- (d) `main()` વિધેયની ઉપર ઘોષિત કરેલ ચલને _____ ચલ તરીકે ઓળખવામાં આવે છે.

9. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો:

- (a) C ભાષાનો પ્રોગ્રામ માત્ર એક કિંમત મોકલી (pass કરી) શકે છે.
- (b) વિધેય એક સાથે અનેક સંલગ્ન મૂલ્યો (arguments) મોકલી શકે નહીં.
- (c) `main()` વિધેયમાંથી એકનું એક ઉપયોગકર્તા નિર્મિત વિધેય અનેકવાર બોલાવી શકાય નહીં.
- (d) C ભાષાના પ્રોગ્રામમાં `main()` પોતે પણ એક પ્રકારનું વિધેય જ છે.
- (e) લાઈબ્રેરી વિધેય માટે પ્રતિકૃતિ (function prototype)ની ઘોષણા પ્રોગ્રામની શરૂઆતમાં કરવી જરૂરી નથી.

10. નીચેના પ્રશ્નો માટે આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ લખો :

(1) C ભાષામાં `main()` વિધેય એ

- (a) ઉપયોગકર્તાનિર્મિત વિધેય છે.
- (b) લાઈબ્રેરી વિધેય છે.
- (c) ચાવીરૂપ શબ્દ છે.
- (d) અનામત વિધેય છે.

(2) સામાન્ય રીતે main() વિધેય શું પરત કરે છે?

- (a) Char કિંમત (b) Float કિંમત
(c) Int કિંમત (d) Double કિંમત

(3) જ્યારે વિધેય કોઈ પણ કિંમત પરત ન કરતું હોય ત્યારે return type તરીકે કયા ચાવીરૂપ શબ્દનો ઉપયોગ કરાય છે?

- (a) int (b) main (c) void (d) auto

(4) નીચેનામાંથી કયું આંતરપ્રસ્થાપિત લાઈબ્રેરી વિધેય નથી?

- (a) pow() (b) printf() (c) sum() (d) sqrt()

(5) નીચેના પ્રોગ્રામના અમલ પછી number2 નામના ચલની કિંમત શું મળશે?

```
int main(){  
    int number1 = 2, number2;  
    number2 = sqr(number1);  
    printf("number2 = %d",number2);  
    return 0;  
}
```

```
int sqr(int n){  
    return(n*n);  
}  
//End of program
```

- (a) number2 = 4 (b) number2 = 6
(c) number2 = 8 (d) number2 = 18

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો પાર પાડવા માટે C ભાષામાં પ્રોગ્રામ તૈયાર કરો :

1. main() વિધેયમાં N નામનો પૂર્ણાંક મેળવો. N નામના ચલની કિંમતને સંલગ્ન મૂલ્ય (argument) તરીકે વિધેયને મોકલો. વિધેયનો ઉપયોગ કરી આપેલ N સંખ્યા માટે નીચે પ્રમાણેની હારમાળા(series)નો સરવાળો દર્શાવો.
$$1 + 2 + 3 + 4 + \dots + N$$
2. એક એવું ઉપયોગકર્તાનિર્મિત વિધેય તૈયાર કરો જે કોઈ સ્વાગત સંદેશ (welcome message) દસ વાર દર્શાવે.
3. એક એવું વિધેય તૈયાર કરો જે xની n ઘાતની ગણતરી કરે.
4. એક એવું ઉપયોગકર્તાનિર્મિત વિધેય તૈયાર કરો જે આપેલી બે સંખ્યા પૈકી મોટી સંખ્યા દર્શાવે.
5. main() વિધેયમાંથી એક સંખ્યા વાંચો. તેને ઉપયોગકર્તાનિર્મિત વિધેય તરફ સંલગ્ન મૂલ્ય તરીકે તબદીલ કરો અને દર્શાવો કે આપેલ સંખ્યા ધન સંખ્યા છે કે ઋણ સંખ્યા છે?
6. તમારા પ્રોગ્રામમાં એક સંખ્યા વાંચો, તેના ઉપર લાઈબ્રેરી વિધેય abs(), sin(), cos() અને log()નો અમલ કરો અને તેનું પરિણામ દર્શાવો.
7. ઉપયોગકર્તા પાસેથી કેરેક્ટર એરે (string) મેળવો અને મળેલ અક્ષરમાળા(string)ના અક્ષરોને ઊંધા ક્રમમાં દર્શાવો.



પરિશિષ્ટ I



દશાંકી સંખ્યાની તેને સંલગ્ન દ્વિઅંકી, અષ્ટાંકી અને સોળઅંકી સંખ્યાઓ સાથેની યાદી.

દશાંકી સંખ્યા (0,1,2,...,9)	દ્વિઅંકી સંખ્યા (0,1)	અષ્ટાંકી સંખ્યા (0,1,2,...,7)	સોળઅંકી સંખ્યા (0,1,2,...,9,A,B,C,D,E,F)
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14



પરિશિષ્ટ II



અક્ષરોની આસ્કી (ASCII) કિંમતો

ASCII		ASCII		ASCII		ASCII	
કિંમત	અક્ષર	કિંમત	અક્ષર	કિંમત	અક્ષર	કિંમત	અક્ષર
000	NUL	032	blank	064	@	096	
001	SOH	033	!	065	A	097	a
002	STX	034	"	066	B	098	b
003	ETX	035	#	067	C	099	c
004	EOT	036	\$	068	D	100	d
005	ENQ	037	%	069	E	101	e
006	ACK	038	&	070	F	102	f
007	BEL	039	'	071	G	103	g
008	BS	040	(072	H	104	h
009	HT	041)	073	I	105	i
010	LF	042	*	074	J	106	j
011	VT	043	+	075	K	107	k
012	FF	044	,	076	L	108	l
013	CR	045	-	077	M	109	m
014	SO	046	.	078	N	110	n
015	SI	047	/	079	O	111	o
016	DLE	048	0	080	P	112	p
017	DC1	049	1	081	Q	113	q
018	DC2	050	2	082	R	114	r
019	DC3	051	3	083	S	115	s
020	DC4	052	4	084	T	116	t
021	NAK	053	5	085	U	117	u
022	SYN	054	6	086	V	118	v
023	ETB	055	7	087	W	119	w
024	CAN	056	8	088	X	120	x
025	EM	057	9	089	Y	121	y
026	SUB	058	:	090	Z	122	z
027	ESC	059	;	091	[123	{
028	FS	060	<	092	\	124	
029	GF	061	=	093]	125	}
030	RS	062	>	094	?	126	~
031	US	063	?	095	-	127	DEL

33 થી 126 અક્ષરો મુદ્રણ થઈ શકે તે પ્રકારના (printable) છે. અન્ય અક્ષરો નિયંત્રણ અક્ષરો છે, જે મુદ્રિત કરી શકતા નથી.



પરિશિષ્ટ III



સી પ્રક્રિયકોનો સારાંશ

પ્રક્રિયક	પ્રક્રિયાનો ઉપયોગ	સંબંધિતતા	અગ્રતા
()	Function call	Left to Right	First
[]	Array expression		
→	Structure operator		
.	Structure operator		
+	Unary plus	Right to Left	Second
-	Unary minus		
++	Increment		
--	Decrement		
!	Logical NOT		
~	Bitwise NOT		
*	Pointer operator		
&	Address operator		
sizeof()	Size of operand		
*	Multiplication	Left to Right	Third
/	Division		
%	Modulo division		
+	Binary addition	Left to Right	Fourth
-	Binary subtraction		
<<	Left shift	Left to Right	Fifth
>>	Right shift		
<	Less than	Left to Right	Sixth
<=	Less than equal to		
>	Greater than		
>=	Greater than equal to		
==	Equal to	Left to Right	Seventh
!=	Not equals to		
&	Bitwise AND	Left to Right	Eighth
^	Bitwise XOR	Left to Right	Ninth
	Bitwise OR	Left to Right	Tenth
&&	Logical AND	Left to Right	Eleventh
	Logical OR	Left to Right	Twelfth
?:	Conditional operator	Right to Left	Thirteenth
=, *, -, &=, +=,		Right to Left	Fourteenth
^=, =, <<=, >>=	Assignment Operator		
,	Comma operator	Left to Right	Fifteenth



પરિશિષ્ટ IV



નિયમિત ઉપયોગમાં લેવાતી કેટલીક હેડર ફાઈલ

આપણે જાણીએ છીએ કે, સી ભાષામાં વિધેયો ઘણા જ મહત્વના છે. વિવિધ પ્રક્રિયાઓના અમલ માટે સી ભાષા આંતરસ્થાપિત (inbuilt) વિધેયોનો સંગ્રહ ધરાવે છે. આ વિધેયોનો સંગ્રહ હેડર ફાઈલમાં જૂથ બનાવીને કરવામાં આવ્યો છે. આવી હેડર ફાઈલોના સંગ્રહને સી લાઈબ્રેરી કહે છે. પ્રોગ્રામર ઉપયોગમાં લઈ શકે તે પ્રકારની કેટલીક હેડર ફાઈલોની યાદી નીચે આપવામાં આવી છે.

ફાઈલનું નામ	ઉદ્દેશ
<stdio.h>	પ્રમાણભૂત નિવેશ/નિર્ગમ માટેના વિધેયો
<ctype.h>	અક્ષરની ચકાસણી અને રૂપાંતરણ માટેના વિધેયો
<math.h>	ગાણિતિક વિધેયો
<stdlib.h>	અક્ષરોની હારમાળા (સ્ટ્રિંગ)નું રૂપાંતરણ, મેમરીની ફાળવણી અને યાદચ્છિક અંકોના નિર્માણ માટે ઉપયોગમાં લેવામાં આવતા વિવિધ પ્રકારના વિધેયો
<string.h>	અક્ષરોની હારમાળા (સ્ટ્રિંગ)ને લગતા વિધેયો

ઉપરોક્ત હેડર ફાઈલમાં સમાવવામાં આવેલા કેટલાક વિધેયો નીચે દર્શાવવામાં આવ્યા છે. વિધેયના ઉપયોગ માટે નિવેશ (input) જરૂરી છે. ઈનપુટ એ વિધેયની વ્યાખ્યાનો એક ભાગ છે અને તેને આર્ગ્યુમેન્ટ (argument) તરીકે ઓળખવામાં આવે છે. આર્ગ્યુમેન્ટના ડેટા પ્રકાર જુદા જુદા હોઈ શકે છે. આર્ગ્યુમેન્ટને ઓળખવા માટે નીચે આપેલ અક્ષરોના કોડનો ઉપયોગ કરવામાં આવે છે.

c	- char પ્રકારની આર્ગ્યુમેન્ટ
d	- double પ્રકારની આર્ગ્યુમેન્ટ
f	- file પ્રકારની આર્ગ્યુમેન્ટ
i	- int પ્રકારની આર્ગ્યુમેન્ટ
l	- long પ્રકારની આર્ગ્યુમેન્ટ
p or *	- pointer પ્રકારની આર્ગ્યુમેન્ટ
s	- string પ્રકારની આર્ગ્યુમેન્ટ
u	- unsigned પ્રકારની આર્ગ્યુમેન્ટ

કેટલાક વિધેયો જુદા-જુદા પ્રકારની કિંમતો પરત કરવાની ક્ષમતા પણ ધરાવે છે. જુદા જુદા વિધેયો, તેની આર્ગ્યુમેન્ટ અને પરત કિંમતના પ્રકાર વિશે નીચે મુજબ વિગતો આપવામાં આવી છે.

Functions in <stdio.h> file

વિધેય	પરત કિંમત	સમજૂતી
getc(f)	int	ફાઈલ f માંથી એક અક્ષર ઉમેરવા માટે
getchar(void)	int	પ્રમાણભૂત ઈનપુટ સાધનની મદદથી એક અક્ષર ઉમેરવા માટે
gets(s)	char*	પ્રમાણભૂત ઈનપુટ સાધનની મદદથી અક્ષરોની હારમાળા (સ્ટ્રિંગ) ઉમેરવા માટે
printf(...)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ વિગતો મોકલવા માટે.
putc(c,f)	int	ફાઈલ f માં એક અક્ષર મોકલવા માટે.
putchar(c)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ એક અક્ષર મોકલવા માટે

puts(s)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ અક્ષરોની હારમાળા (સ્ટ્રિંગ) મોકલવા માટે
scanf(...)	int	પ્રમાણભૂત ઇનપુટ સાધનની મદદથી વિગતો ઉમેરવા માટે.

<ctype.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
isalnum(c)	int	આર્ગ્યુમેન્ટ અક્ષર કે અંક પ્રકારની (alphanumeric) છે કે નહીં તે ચકાસવા માટે. True માટે શૂન્યેતર અને False માટે શૂન્ય પરત કરશે.
isalpha(c)	int	આર્ગ્યુમેન્ટ અક્ષર (alphabet) છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isascii(c)	int	આર્ગ્યુમેન્ટ ASCII અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isctrl(c)	int	આર્ગ્યુમેન્ટ ASCII નિયંત્રણ અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isdigit(c)	int	આર્ગ્યુમેન્ટ દશાંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
islower(c)	int	આર્ગ્યુમેન્ટ નાના (small/lower) અક્ષરમાં છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isodigit(c)	int	આર્ગ્યુમેન્ટ અષ્ટાંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isprint(c)	int	આર્ગ્યુમેન્ટ મુદ્રણક્ષમ (printable) ASCII અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
ispunct(c)	int	આર્ગ્યુમેન્ટ વિરામચિહ્ન (punctuation) પ્રકારનો અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isspace(c)	int	આર્ગ્યુમેન્ટ વ્હાઈટ-સ્પેસ પ્રકારનો અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isupper(c)	int	આર્ગ્યુમેન્ટ મોટા (capital/upper) અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isxdigit(c)	int	આર્ગ્યુમેન્ટ સોળાંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
toascii(c)	int	આર્ગ્યુમેન્ટની કિંમતને ASCIIમાં ફેરવવા માટે.
tolower(c)	int	અક્ષરને લોઅર કેસમાં ફેરવવા માટે.
toupper(c)	int	અક્ષરને અપર કેસમાં ફેરવવા માટે.

<math.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
acos(d)	double	d ની આર્ક કોસાઈન કિંમત પરત કરવા માટે.
asin(d)	double	d ની આર્ક સાઈન કિંમત પરત કરવા માટે.
atan(d)	double	d ની આર્ક ટેન્જન્ટ કિંમત પરત કરવા માટે.
atan2(d1, d2)	double	d1/d2 ની આર્ક ટેન્જન્ટ કિંમત પરત કરવા માટે.
ceil(d)	double	આપેલ સંખ્યાથી વધુ હોય તેવી સૌથી નજીકની પૂર્ણાંક સંખ્યા પરત કરવા માટે
cos(d)	double	d ની કોસાઈન કિંમત પરત કરવા માટે
cosh(d)	double	d ની અતિપરવલય (hyperbolic) કોસાઈન કિંમત પરત કરવા માટે
exp(d)	double	d નો e ઘાતાંક પરત કરવા માટે

fabs(d)	double	d ની નિરપેક્ષ કિંમત પરત કરવા માટે
floor(d)	double	આપેલ સંખ્યાથી ઓછી હોય તેવી સૌથી નજીકની પૂર્ણાંક સંખ્યા પરત કરવા માટે
fmod(d1,d2)	double	d1/d2ની શેષ પરત કરવા માટે (d1ની નિશાની સાથે).
labs(1)	long int	1 ની નિરપેક્ષ (absolute) કિંમત પરત કરવા માટે
log(d)	double	d નો કુદરતી લઘુગુણક (natural logarithm) પરત કરવા માટે
log10(d)	double	d નો (10 આધારિત) લઘુગુણક પરત કરવા માટે.
pow(d1,d2)	double	d1 ના d2 ઘાતાંક જેટલી કિંમત પરત કરવા માટે.
sin(d)	double	d ની સાઈન કિંમત પરત કરવા માટે.
sinh(d)	double	d ની અતિપરવલય (hyperbolic) સાઈન કિંમત પરત કરવા માટે.
sqrt(d)	double	d નું વર્ગમૂળ પરત કરવા માટે.
tan(d)	double	d ની ટેન્જન્ટ કિંમત પરત કરવા માટે.
tanh(d)	double	d ની અતિપરવલય (hyperbolic) ટેન્જન્ટ કિંમત પરત કરવા માટે

<stdlib.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
abs(i)	int	i ની નિરપેક્ષ (absolute) કિંમત પરત કરવા માટે.
atof(s)	double	સ્ટ્રિંગને બમણી ચોકસાઈ ધરાવતી (double-precision) કિંમતમાં ફેરવવા માટે
atoi(s)	int	સ્ટ્રિંગને પૂર્ણાંક સંખ્યામાં ફેરવવા માટે.
atol(s)	long	સ્ટ્રિંગને લોંગ પૂર્ણાંક સંખ્યામાં ફેરવવા માટે
calloc(u1,u2)	void*	u1 ઘટકોને u2 બાઈટ લંબાઈ ધરાવતા મેમરીના ખંડ ફાળવે છે. ફાળવણીની શરૂઆતના ખંડનું પોઈન્ટર પરત કરશે.
exit(u)	void	તમામ ફાઈલો અને બફર બંધ કરી પ્રોગ્રામ બંધ કરવા માટે. (પ્રોગ્રામના અંતની સ્થિતિ દર્શાવવા વિધેય દ્વારા u ની કિંમત ઉમેરવામાં આવશે.)
free(p)	void	p થી શરૂઆત થતી હોય તેવા મેમરીના ફાળવેલા ખંડને મુક્ત કરવા માટે.
malloc(u)	void*	u બાઈટ જેટલી મેમરી ફાળવવા માટે ફાળવવામાં આવેલ જગ્યાની શરૂઆતનું પોઈન્ટર પરત કરશે.
rand(void)	int	યાદસ્થિત (random) ધન પૂર્ણાંક પરત કરશે.
realloc(p,u)	void*	પોઈન્ટર ચલ p ને u બાઈટ જેટલી નવી મેમરી ફાળવશે. નવી મેમરી જગ્યાની શરૂઆતનું પોઈન્ટર પરત કરશે.

<string.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
strcmp(s1, s2)	int	બે સ્ટ્રિંગને શબ્દકોશ રચના પ્રમાણે (lexicographically) સરખાવશે. જો s1 < s2 હોય તો ઋણ સંખ્યા, s1 અને s2 સરખા હોય તો 0 અને s1 > s2 હોય તો ધન સંખ્યા પરત કરશે.
strcmpi(s1,s2)	int	બે સ્ટ્રિંગના કેસને અવગણી શબ્દકોશ રચના પ્રમાણે (lexicographically) સરખાવશે. જો s1 < s2 હોય તો ઋણ સંખ્યા, s1 અને s2 સરખા હોય તો 0 અને s1 > s2 હોય તો ધન સંખ્યા પરત કરશે.
strcpy(s1, s2)	char*	s2 સ્ટ્રિંગની નકલ s1માં કરશે.
strlen(s)	int	s સ્ટ્રિંગમાં આવેલ અક્ષરોની સંખ્યા પરત કરશે.

