એકમ-3 પ્રોગ્રામિંગ (Programming)

: માળખું :

3.0 હેતુઓ
3.1 પરિચય
3.2 અલગોરિધમ અને ફ્લોચાર્ટ
3.3 પ્રોગ્રામ વિકાસ જીવન સાયકલ
3.3.1 પ્રોગ્રામ ચોકસાઈકરણ
3.3.2 પ્રોગ્રામ આકાર
3.3.3 પ્રોગ્રામ કોડ
3.3.4 પ્રોગ્રામ તપાસ
3.3.5 પ્રોગ્રામ ડોક્યુમેન્ટેશન
3.3.6 પ્રોગ્રામ જાળવણી
3.4 પ્રોગ્રામિંગનાં કારણો
3.4.1 પ્રક્રિયા ભાષા
3.4.2 પદાર્થ આધારિત પ્રોગ્રામીંગ ભાષા
3.4.3 વિશિષ્ટ હેતુ ભાષા
3.5 પ્રોગ્રામિંગ ભાષાની પેઢીઓ
3.5.1 યંત્ર ભાષા
3.5.2 એસેમ્બલી ભાષા
3.5.3 ઊંચા સ્તરની ભાષા
3.5.4 ચોથી પેઢીની ભાષા
3.5.5 કુદરતી ભાષા
3.6 ઉપસંહાર
3.7 તમારી પ્રગતિ ચકાસોનાં જવાબો
3.8 ચાવીરૂપ શબ્દો
3.9 સંદર્ભો અને આગળનું વાંચન

3.0 હેતુઓ (OBJECTIVES)

પાછલાં એકમમાં તમે કૉમ્પ્યુટર સોફ્ટવેરનાં મૂળભૂત ખ્યાલો વિશે ભણ્યા. આ એકમમાં આપણે પ્રોગ્રામીંગ અને પ્રોગ્રામીંગ ભાષા વિશે સમજીશું.

આ એકમને વાંચ્યા બાદ તમે આ સમજી શકશો.

- પ્રોગ્રામીંગના મૂળભૂત ખ્યાલો સમજાવો.
- પ્રોગ્રામીંગની ભાષાની વિભિન્ન શ્રેણીઓથી પરિચિત થશો.
- પ્રોગ્રામીંગ ભાષાનાં પ્રકારોથી વાકેફ થશો.

3.1 પરિચય (INTRODUCTION)

કૉમ્પ્યુટર તેની પોતાની મેળે વિચારતું નથી કે નિર્ણય લઈ શકતું નથી. કૉમ્પ્યુટરનાં ઉપયોગ વડે સમસ્યાનું સમાધાન કરવા તેને તે સમજી શકે તેવાં સ્વરૂપમાં માહિતીઓની ગોઠવણ આપવી જોઇએ.

આ સૂચનાઓને લખવાની પ્રક્રિયાને પ્રોગ્રામીંગ કહેવાય છે. પ્રોગ્રામીંગ શું છે તેની સમજણ સરળ અને પ્રખ્યાત ઉદા. દ્વારા સમજી શકાય. ધારો કે, શબ્દ માટે શબ્દકોશની શોધ.

આ સમસ્યાનું વાજબી નિરાકરણ એ છે કે શબ્દકોશ લો અને શબ્દનાં પ્રારંભિક અક્ષરોથી પાનાને શોધો. એક વખત શબ્દ મળી જાય, ત્યારબાદ તે શબ્દના બીજા અક્ષરને શોધો, શબ્દકોશમાં આપેલ તેનાં અર્થને વાંચો.

ચાલો ધારીએ કે શબ્દકોશ કૉમ્પ્યુટરમાં છે. શબ્દકોશમાંથી શબ્દ શોધ્યા બાદ આ સમસ્યાનાં નિરાકરણ માટે આપણે પ્રોગ્રામ લખવાની જરૂરિયાત છે. અલ્ગોરીધમ સમસ્યાનું નિરાકરણ નીચે મુજબ કરે છે.

(1) શોધવાનાં શબ્દને તે ઇનપુટ તરીકે લે છે.

(2) જ્યારે સંપૂર્ણ સમાન શબ્દ ન મળે ત્યાર સુધી શબ્દકોશમાં ઉપલબ્ધ શબ્દોને એક પછી એક એમ સરખાવે છે.

(3) ઇનપુટ શબ્દ માટે અર્થને આઉટપુટ તરીકે દર્શાવે છે.

આ નિરાકરણ કુદરતી ભાષામાં થયું જે કૉમ્પ્યુટર સમજી શકતું નથી. તેથી આપણે ભાષાને માહિતીની વ્યવસ્થામાં લખવી જરૂરી છે કે જે કૉમ્પ્યુટર સમજી શકે છે. આવી માહિતીની ગોઠવણ કે જે ક્રમબદ્ધ રીતે ગોઠવાયેલ હોય તે કૉમ્પ્યુટરને સમસ્યા નિરાકરણ માટે માર્ગદર્શન આપે છે, જેને પ્રોગ્રામ કહેવાય છે.

પ્રોગ્રામ લખવાની પ્રક્રિયાને પ્રોગ્રામીંગ કહેવાય છે. તેમાં સમસ્યાનું સમાધાન થાય છે અને તે એવા સ્વરૂપમાં લખે છે કે જે કૉમ્પ્યુટર યંત્ર સમજી શકે છે અને સંપાદિત કરી શકે છે. જ્યારે ધારવામાં આવે કે કેવી રીતે સમસ્યાનું સમાધાન થશે ત્યારે ત્રણ વિવરણ માનવામાં આવે છે તેને લેવા પડે.

- યંત્રમાં કયો માહિતીનો પ્રવાહ છે ?
- યંત્રમાં કયો માહિતીનો પ્રવાહ નથી ?
- યંત્ર આ માહિતી સાથે શું કરે છે ?

કૉમ્પ્યુટર સમસ્યા નિરાકરણ માટે કુદરતી ભાષાઓ જેવી કે અંગ્રેજી અથવા હિન્દીને સમજી શકતું નથી. માહિતી આદાન-પ્રદાન અને કમાન્ડ માટે આપણે પ્રોગ્રામીંગ ભાષાની જરૂર પડે અને પ્રોગ્રામીંગ ભાષા સમજવા માટે, સંજ્ઞા, શબ્દો અને ભાષાનાં નિયમો શીખવા પડે. પ્રોગ્રામીંગ કરવા માટે ઘણી ભાષાઓ ઉપલબ્ધ છે. વિસ્તૃત અમલીકરણો માટે તે વિકસાવવામાં આવેલ છે. આ અમલીકરણોને આધારે પ્રોગ્રામ બે પ્રકારનાં છે : યંત્ર પ્રોગ્રામ અને અમલીકરણ પ્રોગ્રામ.

યંત્ર પ્રોગ્રામ : આ પ્રોગ્રામથી કૉમ્પ્યુટરનો ઉપયોગ ઉપલબ્ધ થાય છે. સંચાલન તંત્ર જેવા કે DOS અને વિન્ડોઝ જેમાં ઘણા અન્ય પ્રોગ્રામો, ઇનપુટ/આઉટપુટ ઘટકનાં નિયંત્રણ માટે, મેમરી, પ્રક્રિયા વગેરે માટે સમાવેશ થાય તે બધા તંત્ર પ્રોગ્રામ છે. યંત્ર સોફ્ટવેરનાં વિકાસ માટે C ભાષાનો વિશાળ રીતે ઉપયોગ થાય છે.

અમલીકરણ પ્રોગ્રામ : અમલીકરણ પ્રોગ્રામ વિસ્તૃત અમલીકરણ, જેવા કે ગ્રંથાલય, સ્વયં સંચાલન, માહિતી સુધારણા, માહિતી પ્રક્રિયા, શોધ નિયંત્રણ વગેરે માટે આકારિત કરવામાં આવેલ છે. આ અમલીકરણો ઇનપુટ મેળવવા, આઉટપુટ ઉત્પાદિત કરવા, ગણતરી કરવા, સંગ્રહ અને ડેટા સુધારણા કરવા માટે સક્ષમ છે. ઘણી બધી ભાષાઓ છે કે જેમાં સૂચનાઓનો સમાવેશ થાય છે, તે આ બધી ક્રિયાઓને સહકાર આપે છે. પ્રોગ્રામીંગ સાથે ચાલુ રહેતા પહેલાં, આપણે અલગોરિધમ અને ફ્લોચાર્ટ શું છે તે જોવું જોઇએ અને તે કેવી રીતે પ્રોગ્રામીંગમાં મદદ કરે છે.

◆ **તમારી પ્રગતિ ચકાસો**

(1) પ્રોસેસ ડેટાને અનુસરતી કૉમ્પ્યુટર માટેની માહિતીની યાદીને તમે શું કહેશો ?

નોંધ : (1) તમારો જવાબ નીચે આપેલ જગ્યામાં લખો.

(2) આ એકમને અંતે આપેલ જવાબ સાથે તમારો જવાબ સરખાવો.

....

....

....

....

....

....

3.2 અલગોરિધમ અને ફ્લોચાર્ટ (ALGORITHM AND FLOW CHARTS)

કૉમ્પ્યૂટર્સ ડેટા અને માહિતીની સમસ્યાનાં નિરાકરણ, પ્રક્રિયા અને સ્વયં-સચાલિત કરવા માટે વપરાય છે. કૉમ્પ્યુટર તે વિવિધ પ્રોગ્રામિંગ ભાષા જેવી કે C ++ અથવા ઘણી બધી અન્ય દ્વારા મળેલ માહિતી વ્યવસ્થાને આધારે બધું કરે છે. કાર્ય કરવા માટે કૉમ્પ્યુટરને માહિતીનો સેટ અપાય છે જેને પ્રોગ્રામ કહે છે. અલગોરિધમ ભાષામાં પ્રસ્તુત થાય છે, જે પ્રોગ્રામીંગ ભાષા તરીકે જાણીતું છે. આવી રીતે પહેલા નોંધ્યું છે તેમ પ્રોગ્રામ લખવા માટે પ્રોગ્રામિંગ ભાષાનું જ્ઞાન હોવું ખૂબ જ જરૂરી છે. આ ઉપરાંત માહિતીની ગોઠવણ સ્વતંત્ર રીતે પણ કોઈપણ પ્રોગ્રામિંગ ભાષામાં પ્રસ્તુત થઈ શકે જેને અલગોરિધમ કહે છે.

અલગોરિધમની ઘણી બધી પરિભાષાઓ છે. ડ્રોમીના મત મુજબ "અલગોરીધમમાં અર્થગ્રહણની ગોઠવણ અને સરળ અર્થવાળા ચોક્કસ તબક્કાઓ હોય છે કે જે પ્રારંભિક શરતોની વ્યવસ્થા આપે છે, પ્રત્યાયન આઉટપુટને પેદા કરે છે અને ચોક્કસ સમય ઘડે છે."

આગળનાં વિભાગમાં આપણે શબ્દકોષમાં શબ્દનાં અર્થ માટે અલગોરિધમ પ્રક્રિયા વિશે જાણ્યું. આવા પ્રકારની અલગોરિધમ પ્રક્રિયા કૉમ્પ્યુટરનો ઉપયોગ કરી સમસ્યા નિરાકરણ કરે છે. આવી રીતે કૉમ્પ્યુટર વડે સમસ્યાનું નિરાકરણ કરવા, આપણે કૉમ્પ્યુટર વડે ઇનપુટ આઉટપુટ સાથે કૉમ્પ્યુટર પ્રોગ્રામ પૂરો પાડવો જોઇએ. ત્યારબાદ ઇનપુટ પ્રોગ્રામને સ્વીકારે છે અને તેને તેની સૂચના મુજબ વિભાજીત કરે છે અને ત્યારબાદ આઉટપુટ પેદા કરે છે કે જે કૉમ્પ્યુટરમાં સમસ્યાનું નિરાકરણ પેદા કરે છે.

અલગોરિધમ અદ્વિતીય ભાષા અને તરાહમાં લખાયેલ હોય છે. અલગોરીધમ જે ભાષામાં લખાય છે તે તેનું પ્રદર્શન કરે છે. કુદરતી ભાષા જેવી કે અંગ્રેજી અલગોરીધમ લખવામાં ઉપયોગી થઈ શકે. ચાલો એક સામાન્ય સમસ્યા 1 થી 'n' નાં આંકડાઓ ઉમેરીને લઇએ (જો તે 'n' = 4 પછી 1+2+3+4 ઉમેરવામાં આવે)

શરૂઆતમાં '0' ની કેટલીક કિંમત ઉમેરો અને 1 થી ગણો. 'n' ને યુઝરમાંથી ઇનપુટ તરીકે સ્વીકારો. જો 'n' ની કિંમત 0 હોય તો દાખલાની કિંમત તરીકે આપો. જો કિંમત 0 કરતા મોટી હોય તો ઇનપુટ n ની કિંમત ન આવે ત્યાં સુધી ગણો. દરેક વખતે વધારો કર્યા બાદ ગણતરીની કિંમતને વધારો. તે નોંધવું જોઇએ કે આવી નાની સમસ્યા માટે પણ કુદરતી ભાષા બંધાયેલી નથી. વિશાળ સમસ્યાના કેસમાં પ્રોગ્રામીંગ કરતી વખતે કુદરતી અલગોરીધમ ભાષા સમજવી અઘરી પડે છે.

અલગોરીધમને લખવાની પછીની પસંદગી ઔપચારીક ભાષા જેવી કે C, C++ વગેરે છે. પરંતુ આકારનાં પ્રારંભિક ફોર્મમાં કોઇએક ઊંચા અમૂર્ત સ્તરે તેનું લખાણ અને વિચારણા થવી જોઇએ. પ્રોગ્રામીંગ ભાષાનો ઉપયોગ વિરામચિહ્નો અને સિન્ટેક્સના ઉપયોગ પર ભાર મૂકે છે કે જે આકારનાં પહેલાનાં તબક્કે નથી થતું. પ્રોગ્રામિંગ ભાષાનું અલગોરીધમ સિન્ટેક્સ મુક્તું હોવું જોઇએ.

અલગોરીધમને અસરકારક રીતે લખવા માટે કુદરતી અને ઔપચારીક ભાષાની વચ્ચેની ભાષા જરૂરી છે. નોટેશન, જે સ્યૂડો કોડ બોલાય છે અને જે સામાન્ય રીતે અલગોરીધમ લખવા વપરાય છે. તે સાદું વાંચવાલાયક અને તેને વ્યાકરણનાં નિયમ નથી. તેને ખૂબ જ ચોક્કસ બંધારણ હોવાથી, વિધાનનાં સંગઠનને દશ્ય કરવું સહેલું છે. સ્યૂડો કોડ ઘણી પ્રોગ્રામીંગ ભાષામાં હોય છે, અલગોરીધમને પ્રોગ્રામમાં ફેરવવા તે સરળ છે.

અલગોરીધમ માટે ઉપરની સમસ્યા કોડમાં નીચે મુજબ છે.

સ્ટેપ-1 વધારો કરવા માટે 1 થી ગણતરીની શરૂઆત કરો.

સ્ટેપ-2 દાખલાનો સંગ્રહ 0 થી વિવિધ ગણતરીનો પ્રારંભ કરો.

સ્ટેપ-3 દાખલા થયા હોય તેવી ગણતરીની સમસ્યા 'નમ'ની ઇનપુટ કિંમત પ્રાપ્ત કરો.

સ્ટેપ-4 દાખલા સાથે ગણતરી ઉમેરો અને દાખલા માટે નક્કી કરો.

સ્ટેપ-5 પછીની કિંમત માટે ગણતરી વધારો.

સ્ટેપ-6 ગણતરીની ચોક્કસ કિંમત ન મળે ત્યાં સુધી 4 અને 5નાં અન્યને પરિવર્તિત કરો.

અલગોરીધમ ફ્લોચાર્ટનાં ઉપયોગ વડે પણ પ્રસ્તુત થાય છે. ફ્લોચાર્ટ સરળ અને ચોખ્ખી અલગોરીધમ વિશેની સમજણ આપે છે. ફ્લોચાર્ટ દ્વારા અલગોરીધમ વિશેની સમજ સરળ બને છે, જે અલગોરીધમ ટેકસ્યુઅલ પ્રદર્શનને સમાન છે.

ફ્લોચાર્ટ માટે નિશાનીઓ :

પ્રોગ્રામીંગની મૂળભૂત પ્રક્રિયા માટે ફ્લોચાર્ટને ઘણી નિશાનીઓ છે. ફ્લોચાર્ટની નિશાનીઓ આ મુજબ છે.

(1) ટર્મિનલ

આ સામાન્ય રીતે પ્રોગ્રામમાં ટર્મિનલ પોઇન્ટને પ્રસ્તુત કરાવે છે જેવા કે BEGIN, END, START, STOP.

(2) ઈનપુટ/આઉટપુટ

સમાંતર બાજુનો બનેલો ચતુષ્કોણ ઇનપુટ/આઉટપુટ કાર્યને પ્રસ્તુત કરે છે, જેમ કે પ્રક્રિયા માટે ડેટા ઉપલબ્ધ કરવા અથવા પ્રક્રિયા માહિતીનું સ્કોર્ડીંગ. આ તબક્કો ઇનપુટ સાધનમાંથી નંબરને મેળવેછે.

(3) પ્રક્રિયા

આ ચોરસ પ્રક્રિયા તંત્રને પ્રસ્તુત કરે છે. પ્રક્રિયા ડેટાને ફેરવે અથવા બદલે છે. સામાન્ય રીતે અસાઇમેન્ટ આ નિશાનીઓ દ્વારા પ્રસ્તુત થાય છે.

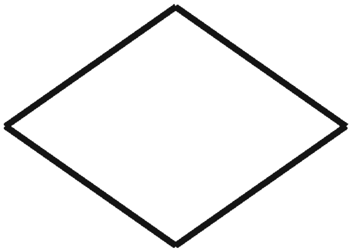
4) પ્રવાહ

લાઇન અથવા એરો પ્રવાહની દિશા-પ્રવાહ નિયંત્રણ પ્રસ્તુત કરે છે. સામાન્ય રીતે પ્રવાહની દિશાએ ડાબીથી જમણી અથવા ઉપરથી નીચે હોય છે.

5) એનોટેશન

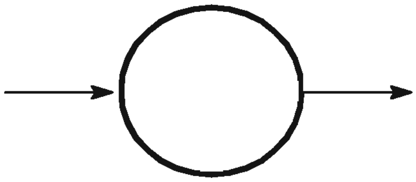
તૂટેલી લાઇન અને ચોરસ એ એનોટેશન કાર્યને પ્રસ્તુત કરે છે - વર્ણનાત્મક ... વધારો અથવા વિસ્તૃત નોંધ.

(6) **નિર્ણય લેતી નિશાની :**



ડાયમંડ એ નિર્ણય અથવા કાર્ય સંચાલનનાં સ્વિચિંગ પ્રકારને પ્રસ્તુત કરે છે. જે ઘણા બધા વિસ્તારોમાંથી કયાં રસ્તાને અનુસર તે નક્કી કરે છે. અવો નિર્ણય દર્શાવે છે જેનો જવાબ હા અથવા ના હોય છે.

7) જોડાણ



ગોળએ જોડાણ નિશાની છે કે જે કાર્યને પ્રવાહમાં પ્રસ્તુત કરવા વપરાય છે. જેમ કે ફ્લોચાર્ટનો ભાગ જે આવતા અથવા અન્ય પેજને સતત કરે છે.

8) પૂર્વ નિર્ધારીત પ્રક્રિયા

બે બાજુનાં ચોરસ નામની પ્રક્રિયાને પ્રસ્તુત કરે છે. જેમાં એક અથવા વધારે સંચાલનો અથવા પ્રોગ્રામીંગ પગલાંઓનો સમાવેશ થાય છે જે બીજે ચોક્કસ થાય છે. જેમ કે મોડ્યૂલ અથવા પેટારૂટીન.

◆ **તમારી પ્રગતિ ચકાસો**

2) અલગોરીધમ અને ફ્લોચાર્ટ વચ્ચે શું તફાવત છે ?

નોંધ : 1) તમારો જવાબ નીચે આપેલ જગ્યામાં લખો.

2) આ એકમને અંતે આપેલ જવાબ સાથે તમારો જવાબ તપાસો.

....

....

....

....

....

....

3.3 પ્રોગ્રામ વિકાસ જીવન સાયકલ (PROGRAM DEVELOPMENT LIFE CYCLE)

પ્રોગ્રામીંગ એ માહિતીની યાદી કરવા માટેની 5 તબક્કાની પ્રક્રિયા છે. સામાન્ય રીતે, પ્રોગ્રામીંગ કૉમ્પ્યુટરમાં રહેલા વિધાનમાં ટાઈપીંગ અથવા કેપીંગનો સમાવેશ કરે છે. આ ઉપરાંત આ પ્રોગ્રામ વિકાસમાં તબક્કાઓ છે. 5 તબક્કાઓ નીચે મુજબનાં છે.

3.3.1 પ્રોગ્રામ ચોકસાઈકરણ (Program Specification)

પ્રોગ્રામીંગ એ માહિતીની યાદી પરિભાષા અથવા પ્રોગ્રામ પૃથક્કરણ પણ કહેવાય છે. આ પગથિયાંમાં પાંચ કાર્યોનાં વિવરણનો સમાવેશ થાય છે. 1) પ્રોગ્રામનાં હેતુઓ, 2) ઇચ્છિત આઉટપુટ, 3) ઇનપુટ ડેટાની જરૂરિયાત, 4) પ્રક્રિયાની જરૂરિયાત, 5) ડોક્યુમેન્ટેશન. ઇચ્છિત વપરાશ કર્તાની જરૂરિયાતને આધારે સંપૂર્ણ આધારિત હોય છે કે જે સોફ્ટવેર યંત્ર એનાલિસ્ટ દ્વારા જરૂરિયાતનાં પૃથક્કરણ દ્વારા પ્રાપ્ત કરી શકાય છે.

◆ **પ્રોગ્રામનાં હેતુઓ**

આપણે એ શીખ્યા કે પ્રોગ્રામીંગમાં સમસ્યાનાં નિરાકરણનો સમાવેશ થાય છે. તેથી આઉટસેટ પર ચોખ્ખું વિધાન બનાવું જરૂરી છે કે કઈ સમસ્યાના નિરાકરણ માટે તમે પ્રયત્ન કરો છે. ઉદા. "વિવિધ ક્ષેત્રો દ્વારા શોધાયેલા ઐતિહાસિક ડેટાબેઝનો વિકાસ કરવો."

◆ **ઇચ્છિત આઉટપુટ**

ઇનપુટ પહેલા આઉટપુટનું વિવરણ કરવું હંમેશા સારું હોય છે. કૉમ્પ્યુટર યંત્ર દ્વારા જરૂરી આખરી પરિણામ નોંધવું જરૂરી છે.

◆ **ઇનપુટ ડેટા**

એકવખત આઉટપુટ જણાઈ જાય, ઇનપુટ ડેટા નિર્ધારીત અને ઓળખી શકાય છે.

◆ **પ્રક્રિયા જરૂરિયાત**

આ તબક્કામાં પ્રક્રિયા કાર્ય કે જે ઇનપુટ ડેટા માટે થાય છે, તે ફરજિયાતપણે આઉટપુટમાંથી પસાર થતા, ચોક્કસ થાય છે.

◆ **પ્રોગ્રામ વિવરણ આકાર**

પ્રોગ્રામીંગ ચાલું રહેત ડોક્યુમેન્ટેશન જરૂરી છે. બધી જ પ્રવૃત્તિઓમાં જેવી કે પ્રોગ્રામ હેતુઓ, ઇચ્છિત આઉટપુટ, ઇનપુટ જરૂરી છે અને જરૂરી પ્રક્રિયા નોંધવી જોઇએ.

3.3.2 પ્રોગ્રામ આકાર (Program Design)

પ્રોગ્રામ વિવરણ પછી પ્રોગ્રામ આકાર આવે છે. અહીંયા તમે બંધારીત પ્રોગ્રામીંગ પ્રયુક્તિનો ઉપયોગ કરી નિરાકરણનું આયોજન કરી શકો છો. આ પ્રયુક્તિઓમાં ટોપ-ડાઉન પ્રોગ્રામ, આકાર, સ્યુડો કોડ, ફ્લોચાર્ટ અને તાર્કિક બંધારણનો સમાવેશ થાય છે.

હવે પ્રોગ્રામનાં આઉટપુટ અને ઇનપુટ ઓળખાય છે, ટોપ ડાઉન પ્રોગ્રામ ડિઝાઇન પ્રોગ્રામનાં પ્રક્રિયાના પગથિયાને ઓળખવામાં ઉપયોગી છે. આવાં પગથિયાંઓ પ્રોગ્રામ મોડલ્યુસ તરીકે ઓળખાય છે. દરેક મોડ્યુલ તાર્કિક સંબંધિત પ્રોગ્રામ વિધાનનું બનેલું છે. સ્યુડો કોડ તમે લખો તે પ્રોગ્રામનાં તર્કની આઉટલાઇન છે. તે પ્રોગ્રામને લખો તે પહેલા તેનો ઉપસંહાર કહો તેવું છે. ફ્લોચાર્ટ એ પાછળનાં એકમમાં ચર્ચાવામાં આવેલ છે. અંતે તાર્કિક બંધારણમાં વિધાન હારમાળાની શ્રેણી, પસંદગી અને લૂપનો સમાવેશ થાય છે.

3.3.3 પ્રોગ્રામ કોડ (Program Code)

પ્રોગ્રામને લખવો તેને કેડિંગ કહેવાય છે. અહીંયા પ્રોગ્રામ આકાર પગથિયા દ્વારા વિકસાવેલ તર્કનો હકીકતમાં પ્રોગ્રામ લખવામાં ઉપયોગ થાય છે. પ્રોગ્રામ કૉમ્પ્યુટરને શું કરવું તેની સૂચના આપે છે. કેડીંગ એ જ્યારે લોકો પ્રોગ્રામીંગ વિશે વિચારતા હોય ત્યારે તેનાં વિશે વિચારે છે. પહેલા નોંધ્યું છે તેમ, પ્રોગ્રામીંગ પ્રક્રિયાનાં માત્ર 5 પગથિયાં છે.

3.3.4 પ્રોગ્રામ તપાસ (Program Testing)

ડિલગીંગ એ પ્રોગ્રામરનો શબ્દ છે જે ભૂલ તપાસ અને સુધારણા માટે વપરાય છે. તેનો અર્થ એક પ્રોગ્રામને કૉમ્પ્યુટરમાં ચલાવવો અને ત્યારબાદ કામ ન કરતા વિભાગમાં બેસાડવો. પ્રોગ્રામીંગ ભૂલો બે પ્રકારની હોય છે. વાક્યરચનાની ભૂલો અને તાર્કિક ભૂલો.

વાક્યરચનાની ભૂલો એ પ્રોગ્રામીંગ ભાષાનાં નિયમો માટે હિંસક છે. ઉદા. C++માં દરેક વિધાન સેમીકોલન સાથે પુરું થાય છે, જો સેમીકોલન વાપરવામાં ન આવે, તો પ્રોગ્રામ વાક્ય રચનાની ભૂલને લીધે ચાલતો નથી. પ્રોગ્રામર જ્યારે ખોટી ગણતરી કરે અથવા પ્રોગ્રામીંગ પ્રક્રિયાને દોડે ત્યારે તાર્કિક ભૂલો થાય છે. ઉદા. પેરોલ પ્રોગ્રામ કે જે વધારાનાં કલાકોની ગણતરી કરતો નથી. તેને તાર્કિક ભૂલો થાય છે.

વાક્યરચનાની અને તાર્કિક ભૂલો તે શોધી અને દૂર કરવા ઘણી બધી પદ્ધતિઓ વપરાય છે, જે આ મુજબ છે.

- **ડેસ્ક તપાસ :** ડેસ્ક તપાસમાં, પ્રોગ્રામનું પ્રૂફ વાંચનની બધી પદ્ધતિઓ દરેક લાઇન વાંચી ભૂલો શોધે છે.
- **સેમ્પલ ડેટા સાથે જાતે તપાસ :** કેટાક્યુલેટર અને સેમ્પલ ડેટાનો ઉપયોગ કરી પ્રોગ્રામર દરેક પ્રોગ્રામ વિધાનને અનુસરે છે અને બધી ગણતરીનું કાર્ય કરે છે. પ્રોગ્રામીંગ તાર્કિક ભૂલો જોઇને પ્રોગ્રામર મેન્યુઅલ ગણતરી કિંમતને કિંમત સાથે સરખાવે છે.
- **અનુવાદમાં પ્રયત્ન :** અનુવાદ પ્રોગ્રામનો ઉપયોગ કરી, પ્રોગ્રામ કૉમ્પ્યુટર વડે ચાલે છે. અનુવાદક લખેલા પ્રોગ્રામને પ્રોગ્રામીંગ ભાષામાંથી યંત્ર ભાષામાં અનુવાદ કરે છે. પ્રોગ્રામ ચાલે તે પહેલા તે વ્યાકરણ રચના ભૂલોથી મુક્ત હોવી જોઇએ. આવી ભૂલો અનુવાદક પ્રોગ્રામ દ્વારા ઓળખવામાં આવે છે.
- **કૉમ્પ્યુટરમાં સેમ્પલ ડેટાની તપાસ :** વ્યાકરણની બધી ભૂલોને સુધાર્યા બાદ પ્રોગ્રામને તાર્કિક ભૂલો માટે તપાસવામાં આવે છે. સેમ્પલ ડેટા, દરેક પ્રોગ્રામ વિધાનના સાચા સંપાદનને તપાસવા વપરાય છે.
- **વપરાશકર્તાના પસંદગી સંગઠન દ્વારા તપાસ :** આને કેટલીવાર બીટા તપાસ કહેવાય છે. તે સામાન્ય રીતે પ્રોગ્રામને તપાસવાનું અંતિમ તબક્કો ગણાય છે. વપરાશકર્તા પ્રોગ્રામ પર પ્રયત્ન કરે છે અને પ્રતિભાવો પૂરા પાડે છે.

3.3.5 પ્રોગ્રામ ડોક્યુમેન્ટેશન (Program Documentation)

ડોક્યૂમેન્ટેશનમાં લેખિત વર્ણનો અને પ્રોગ્રામ વિશેની પ્રક્રિયાઓ અને તેનો કેવી રીતે ઉપયોગ કરવો તેનો સમાવેશ થાય છે. તે પ્રોગ્રામીંગ પ્રક્રિયાના અંતમાં નથી થતું. પ્રોગ્રામીંગ ડોક્યુમેન્ટેશન બધા પ્રોગ્રામીંગ તબક્કા દરમ્યાન થાય છે. આ ડોક્યુમેન્ટેશન પરંપરાગત રીતે પ્રોગ્રામ સાથે થાય છે અને તે છાપેલા દસ્તાવેજમાં હોય છે. આ તબક્કામાં બધા ડોક્યૂમેટેશનનું ફરી વખત અવલોકન અને ચોક્કસ થાય છે. ડોક્યૂમેન્ટેશન, વપરાશકર્તા, સંચાલક અને પ્રોગ્રામર માટે જરૂરી છે કે તેઓ ભવિષ્યમાં પ્રોગ્રામ સાથે જોડાયેલા હોય છે.

3.3.6 પ્રોગ્રામ જાળવણી (Program Maintenance)

અંતિમ તબક્કો પ્રોગ્રામ જાળવણીનો છે. જાળવણીનો હેતુએ દર્શાવે છે કે વર્તમાન પ્રોગ્રામનું ભૂલ રહિત, અસરકારક અને આકર્ષક છે. આ ક્ષેત્રોની પ્રવૃત્તિઓ બે વર્ગમાં વહેંચવામાં આવે છે. સંચાલન અને જરૂરિયાતમાં બદલાવ.

સંચાલન પ્રવૃત્તિઓ સ્થાપન અને સંચાલન ભૂલ સુધારણા, પ્રોગ્રામને માટે સરળ અને બંધારીત પ્રોગ્રામીંગ પ્રયુક્તિના ઉપયોગ દ્વારા ઉત્તમ સોફ્ટવેર સાથે જોડાયેલ છે. સંપૂર્ણ આકારિત પ્રોગ્રામો માટે આ પ્રવૃત્તિઓ નાની હોવી જોઇએ. જરૂરિયાતનો બદલાવ બિનઅવરોધક છે. બધી સંસ્થાઓ વધારા સમયને બદલે છે અને તેનાં પ્રોગ્રામો તેની સાથે બદલાવાં જોઇએ.

◆ **તમારી પ્રગતિ ચકાસો**

3) પ્રોગ્રામિંગનાં 5 પગથિયા કયાં છે ? પ્રોગ્રામનાં વિકાસમાં કોડીંગ શું ભાગ ભજવે છે ?

નોંધ : 1) તમારો જવાબ નીચે આપેલ જગ્યામાં લખો.

2) આ એકમને અંતે આપેલ જવાબ સાથે તમારો જવાબ તપાસો.

....

....

....

....

....

....

3.4 પ્રોગ્રામિંગનાં કારણો (TYPES OF PROGRAMMING)

પાછલા એકમમાં આપણે ભણ્યા કે પ્રોગ્રામિંગ એ માહિતીની ગોઠવણનું લખાણ છે. જો તે પહેલા કરતાં સરળ હોય તો શા માટે આપણે પ્રોગ્રામિંગ ભાષાનો ઉપયોગ નથી કરતા ? પ્રોગ્રામીંગ ભાષા ચોક્કસ જરૂરિયાત પ્રાપ્ત કરવા આકારિત કરવા આવેલ છે. સદ્દભાગ્યે કોઈ એક ભાષા બંધારણીય સમસ્યા નિરાકરણ, જટિલ વહેંચાણ અહેવાલ લખવા, ડેટાબેઝ સાથે આદાનપ્રદાન વેબપેજ આકારિત કરવામાં વપરાય છે. તેથી પ્રોગ્રામીંગ કાર્યનં શુદ્ધિકરણ સાથે પ્રોગ્રામીંગ ભાષાનું શુદ્ધિકરણ પણ હોય છે. ચાલો પ્રોગ્રામિંગ ભાષાના વિવિધ પરિવારો જોઈએ અને પ્રોગ્રામીંગ ભાષાનાં પ્રકારો પણ તેમાં આપેલાં છે.

3.4.1 પ્રક્રિયા ભાષા (Procedural Languages)

ભાષા કે જેમાં પ્રોગ્રામ વિધાનનાં અનુક્રમ મુજબ લખાય. જે ડેટા વસ્તુને વિસ્તારે અને મેમરી સેલનાં વિષય-વસ્તુને ફેરવે તેને પ્રક્રિયા ભાષા કહેવામાં આવે છે. એ આદેશાત્મક ભાષાઓ તરીકે ઓળખાય છે. આ ભાષાનું મૂળભૂત કાર્ય ડેટા કિંમતોનો સંગ્રહ અને સુધારણા છે. ઉદા.

a=1

આ વિધાન કિંમત 1 નો સંગ્રહ aનાં સ્થાને કરે છે.

c = a+b

આ વિધાન a અને b નો સુધારો કરે છે, c માં પરિણામ ને ઉમેરે છે અને સંગ્રહ કરે છે. કેટલીક પ્રક્રિયા ભાષા નીચે મુજબ વર્ણવામાં આવેલ છે.

◆ **FORTRAN**

FORTRAN નામ ફોર્મ્યુલા અનુવાદ(Formula Transalation) માંથી ઉતારવામાં આવેલ છે. તેનું નામ જ સૂચવે છે કે વ્યવસાયિક તરાહ 1957માં પ્રસ્તુત થઈ આને FORTRAN ને પ્રથમ ઊંચા સ્તરની પ્રોગ્રામીંગ ભાષા બનાવી છે. FORTRANને કેટલાંક લક્ષણો છે જે આદર્શ રીતે, ગાણિતીક અથવા કૉમ્પ્યુટર આધારિત સમસ્યા હોય તેને યોગ્ય છે. ઊંચા સ્તરની ભાષા મૂળભૂત રીતે અંગ્રેજી જેવી કે કૉમ્પ્યુટર ભાષા કે જ્યાં આપણે ઊંચા સ્તરની, એસેમ્બલી અને યંત્ર સ્તરની ભાષા મૂળભૂત રીતે સમજવા વિકસાવીએ. પ્રોગ્રામીંગ ભાષા જેવી કે COBOL, PASCAL, C, BASIC વગેરે આ વર્ગમાં આવે છે.

◆ **COBOL**

COBOL નામ સામાન્ય ઉદ્યોગ આધારિત ભાષા પરથી ઉમટી આવેલ છે. COBOLને ઉદ્યોગની જરૂરિયાતો જેવી કે સંચાલન રોકાણ અને પેરોલ માટે આકારિત કરવામાં આવેલ છે. આવા અમલીકરણમાં ઉપસંહાર અહેવાલએ મહત્ત્વનું આઉટપુટ છે. ડેટા ફાઈલનું નિયમન પણ તેમાં સમાવિષ્ટ છે. ઉદ્યોગ વિશ્વમાં ઘણા બધા માસ્ટર ફાઈલને ટ્રાન્મીક્શન ફાઈલમાં બદલાવવાનું કરે છે. ઉદા. માસ્ટર ફાઈલમાં નામ, ઉત્પાદકર્તા અને વિવિધ વસ્તુઓ પ્રાપ્ત જથ્થો; ટ્રાન્ઝીક્શન ફાઈલમાં નામ, જથ્થો અને સમયનાં તબક્કામાં વેચાયેલી વસ્તુની યાદીનો સમાવેશ થાય છે. માસ્ટર ફાઈલ ટ્રાન્ઝીક્શન ફાઈલમાંથી અઠવાડિયે અથવા દરરોજ નવા પ્રાપ્ત જથ્થા અને ઉપસંહાર અહેવાલ નકલ પ્રત્યાઘાત માટે અપડેટ થાય છે.

◆ **PASCAL**

પાસ્કલાઈન કેલ્ક્યુલેટરનાં કેટલાક બ્લેઈઝ પાસ્કલનાં નામ પરથી પ્રોગ્રામીંગ ભાષા પાસ્કલનું નામ પડેલ છે.

પાસ્કલનો આકાર એવો છે કે તે સહેલાઈથી શીખી શકાય છે અને સારી પ્રોગ્રામિંગ પ્રયુક્તિનો ઉપયોગ થાય છે. પાસ્કલ સ્યુડોકોડને મળતી આવે છે. તેથી તે વાંચવામાં સરળ છે અને વાક્ય રચના શીખવામાં સરળ છે. તેમ છતાં પાસ્કલ લેંગ્વેજ, વ્યવસાયિક ભાષા તરીકે વધારે ઉપયોગી નથી, પરંતુ પ્રોગ્રામીંગ

વાતાવરણ જેવા કે ડેલ્ફી પાસ્કલ ભાષાનો વિષે આધારિત આધુનિક ગ્રાફિકલ ઇન્ટરફેસ સાથેનો વિકાસની સુવિધા સાથે ઉપયોગ કરે છે.

◆ **C**

ડેનીસ રિચી દ્વારા 1970ની શરૂઆત AT & T બેલ લેબોરેટરીમાં C વિકસાવવામાં આવી. તે વાસ્તવિક રીતે યંત્ર પ્રોગ્રામિંગ માટે વિકસાવવામાં આવી હતી. ખાસ કરીને UNIX સંચાલન તંત્ર લખાણ માટે છે. તે બે કારણો માટે સામાન્ય હેતુસર પ્રખ્યાત બની પેલું UNIX સાથેનો સંબંધ, બીજું કારણ તેની અસરકારકતા, ગતિ કે જે સંચાલનને સંપાદિત કરે છે. અસરકારકતા એ વાસ્તવિક્તામાંથી ઉતરેલ છે કે C પ્રોગ્રામ નીચા સ્તરની માહિતીનો ઉપયોગ કરે છે જેવી કે મેમરીમાં ડેટા સંગ્રહાયેલા છે તેનું જ્ઞાન, તેનું શક્તિશાળી વિધાન છે. અને યંત્રમાં ઊંચા સ્તરની ભાષાની શક્યતા સમાયેલી છે.

C નું મજબૂત લક્ષણ એ છે કે તે ડેટા-પ્રકાર આપે છે જેને પોઇન્ટર કહેવાય છે. પોઇન્ટર પ્રકારની વિવિધતા, ઇન્ટીગર્સને બદલે મેમરી સરનામા, સાચા નંબર અને અક્ષરોનો સમાવેશ કરે છે. ઉદા. વિધાન

```
int*    intpainter
```

દર્શાવે છે કે inpainter પોઇન્ટર તરીકે યોગ્ય છે કે જે મેમરી સરનામા સ્થાન અને ઇન્ટીગેટ ડેટાનો સમાવેશ કરે છે.

```
int     A=3;
intpainter = &A
```

પ્રથમ વિધાન યોગ્યતાને દર્શાવે છે જેને 'A' બોલાય છે અને તેની કિંમત '3' નક્કી કરે છે. બીજુ વિધાન ઈન્ટપોઈન્ટર પોઈન્ટીંગ 'A' કરે છે.આપણે વિધાન દ્વારા A યોગ્ય માટે '10' ની કિંમત નક્કી કરી શકીએ.

```
*intpainter = 10;
```

પોઈન્ટરનો વધારે ઉપયોગ લખાણ તંત્ર પ્રોગ્રામીંગ, સંચાલન તંત્ર, એસેમ્બલર્સ અને પ્રોગ્રામમાં થાય છે જે કૉમ્પ્યુટરને ઇનપુટ/આઉટપુટ સાધનો સાથે આદાન-પ્રદાનની સુવિધા આપે છે. ઉદા. ધારો કે PC પર માઉસ માટે ડિવાઇસ ડ્રાઇવરમાં લખવાની સમસ્યા છે. માઉસ જ્યાં જોડાયેલ હોય ત્યાં વોલ્ટેજ સ્તર દ્વારા માન્સની સ્થિતિ બદલો. તે મેમરીનાં ચોક્કસ સ્થાને વોલ્ટેજ સ્તરને સંગ્રહ કરે છે. C આ મેમરીમાં પ્રક્રિયા પૂરી પાડે છે.

તેમ છતાં C યંત્ર પ્રોગ્રામીંગ માટે ખૂબ જ ઉપયોગી તેમજ સામાન્ય હેતુ પ્રોગ્રામીંગ માટે પણ ખૂબ ઉપયોગી છે.

◆ **ADA**

Ada સામાન્ય ઊંચા સ્તરની ભાષા માટે વિકસાવવામાં આવેલ હતી જે અમેરીકાની ઓર્મીસેવાના સંરક્ષણ કરારો માટે જરૂરી હતી. Ada ને C++ ની જેમ વિશાળ ભાષા છે અને તે માત્ર સંરક્ષણ ઉદ્યોગ દ્વારા અપનાવવામાં નથી આવેલ પરંતુ અન્ય યંત્ર અમલીકરણ દ્વારા પણ અપનાવેલ છે અને સામાન્ય હેતુ તરીકે સારી ભાષા છે.

Ada તેની મલ્ટી પ્રક્રિયા ક્ષમતા માટે જાણીતી છે. આ ક્ષમતા તેને ઘણા બધા કાર્યો સ્વતંત્ર રીતે કરવાની ક્ષમતા અને સીધું પ્રત્યાયન કરવાની ક્ષમતા આપે છે.

3.4.2 પદાર્થ આધારિત પ્રોગ્રામીંગ ભાષા (Object Oriented Programming Languages)

એલાન કેયનાં કાર્યથી 1970ની શરૂઆતમાં સંશોધન કેન્દ્રમાં પદાર્થ આધારિત પ્રોગ્રામીંગ શરૂ થયું. પરિણામે સ્મોલટોક ભાષા આવી.

પદાર્થ આધારિત પ્રોગ્રામીંગનો ખ્યાલ મુખ્ય કાર્યમાંથી પેટાકાર્યમાં વિભાજીત થયેલ છે. પ્રોગ્રામનો વિશાળ વિધાન તરીકે વિચાર કરવામાં આવે છે. જે મુખ્ય કાર્ય કરે છે. તેમ છતાં મુખ્ય પ્રોગ્રામ સરળતાથી વિવિધ મોડ્યુલ્સ કે જે પેટાકાર્ય કરે છે તેને સંબોધે છે. પદાર્થને સમજવા ચાલો ગ્રંથાલય તંત્રનું ઉદા. લઈએ. પ્રવાહિતતા, પ્રાપ્તિ, સૂચિ વગેરે મુખ્ય પદાર્થ છે. દરેક પદાર્થ ગ્રંથાલય તંત્રમાં કાર્ય કરતું ઉદા. છે.

◆ **JAVA**

નવા પ્રોગ્રામીંગ વૈશ્વિક વેબ સાથે જોડાયેલ છે. નવા પ્રોગ્રામ જેવી રીતે લોકલ તંત્રમાં ડેટા પ્રક્રિયા કરે છે તેમ સરળતાથી વેબમાં પ્રક્રિયા કરે છે. સન માઇક્રો સીસ્ટમ મુજબ જાવા સરળ છે, પદાર્થ આધારિત,

વિભાજિત, મધ્યસ્થ, રોબલ્ટ, સુરક્ષિત, બંધારણીય, પોટેબલ, ઊંચું કાર્ય મલ્ટીહેટર્ડ અને ડાયનામિક ભાષાવાળો છે. મલ્ટી વાંચનને વિવિધ કાર્યો એક સમયે કરવાની જે ક્ષમતા હોય છે. મલ્ટીહેટર્ડ પ્રોગ્રામ જાવા દ્વારા સર્જન પામે છે.

3.4.3 વિશિષ્ટ હેતુ ભાષા (Special Purpose Languages)

પ્રક્રિયા ભાષા કે જે આપણે જોઈ તે ઓછે વતે અંશે સામાન્ય હેતુભાષા છે. ઘણી બધી વિશિષ્ટ હેતુ ભાષા વિશિષ્ટ કાર્ય માટે આકારિત થયેલ છે. અહીંયા આપણે ત્રણ વિશિષ્ટ ભાષાનો ઉલ્લેખ કરીશું જે ખૂબ જ પ્રખ્યાત રીતે વપરાય છે.

◆ **બંધારિત ક્વેરી ભાષા (Structured Quary Language - SQL)**

SQL એ ડેટાબેઝ સાથે ઉપયોગમાં લેવાં આકારિત થયેલ છે કે જે સંબંધિત વાસ્તવિકતાઓ અને માહિતીનો સંગ્રહ છે. ડેટાબેઝ ડેટાને કેન્દ્રીત સંગ્રહે છે. ડેટાબેઝનો વપરાશકર્તા ફરજયાતપણે નવા ડેટા ઉમેરવા અને સંગ્રહિત ડેટા સુધારવા સક્ષમ હોવો જોઇએ. ઉદા. ગ્રંથાલય પત્રકનો ડેટાબેઝમાં પુસ્તક શીર્ષક, લેખક નામ, પ્રકાશન નામ, પ્રકાશન કાર્ય વગેરેનો સમાવેશ થાય છે. કેટલોગના ડેટાબેઝનો વપરાશકર્તા બૂક વિશે માહિતી અને ડેટાબેઝમાં રહેલ પુસ્તક વિશેની માહિતી સુધારવા સક્ષમ હોવો જોઇએ. વધારામાં ડેટાબેઝ વપરાશકર્તાનાં પ્રશ્નોનાં ઉકેલ લાવે છે. ઉદા. ગ્રંથાલય પત્રક ચોક્કસ લેખક અથવા પ્રકાશક દ્વારા પુસ્તકને જાહેર કરે છે. આવી ભૂલો SQLમાં મૂકાય છે. SQLએ ભાષા છે જે ડેટાબેઝની ભૂલોને રાખે છે. નમૂનાનું SQL વિધાન નીચે મુજબ છે.

```
SELECT BOOK _ TITLE PUBLISHER, YEAR
FROM CATALOGUE
WHERE AUTHOR NAME, “GUHA”
```

આ SQL વિધાન બુહા લેખકની વર્ષની વિગતો અને પ્રકાશન સાથે બુક શીર્ષકની સુધારણા કરશે.

પ્રેક્ટીકલ ઓક્સટ્રેશન અને અહેવાલ ભાષા (PERL)

Perl આરબિટરી ટેક્સ્ટ ફાઇલને સ્કેન કરવા આકારિત થયેલ છે જે વિવિધ પ્રકારની માહિતીને તારવે છે જેમાં લખાણ અને ઈચ્છિ અહેવાલ, તારવેલી માહિતીને આધારે હોય તેને તારવે છે. ભાષાની વાક્યરચના C ને આધારિત હોય છે. Perl યોગ્ય માળખામાં સમાન પ્રયુક્તિ વિશાળ સ્કેનીંગ પ્રક્રિયા ચોક્કસ લખાણ માટે ઝડપી બનાવવા માટે કરે છે.

◆ **હાયપર ટેક્સ્ટ માર્કઅપ ભાષા (HTML)**

આ ભાષા HTML દસ્તાવેજ સર્જન માટે વપરાય છે, કે જે જ્યારે વેબ બ્રાઉઝર સાથે અવલોકન કરવામાં આવે તે વેબ પેજ બને. HTML દસ્તાવેજમાં લખાણ કે જે વેબપેજમાં દેખાય છે તેનો સમાવેશ થાય છે, સાથે સાથે અસંખ્ય વિશિષ્ટ અક્ષરો કે જેને ટેગ કહેવાય છે. જે ફોરમેટીંગ પ્રાપ્ત કરે છે, ખાસ અસર અને અન્ય HTML દસ્તાવેજને સંદર્ભ આપે છે. તેનો સમાવેશ થાય છે. ટેગ્સ ખૂણા કૌંસ (< >)માં હોય છે. અને તે ઘણીવાર જોડીમાં હોય છે. અંતિમ ટેગ બીજુ ટેગ જોડીનું શરૂઆતના પ્રથમ ટેગ જેવું લાગે છે.

HTML દસ્તાવેજનું સ્વરૂપ આ મુજબ છે.

```
<html>
<head>
<title> શીર્ષક તરીકે દેખાતું લખાણ </title>
</head>
<body>
<html>
લખાણ જે પેજમાં શરીર તરીકે દેખાય
```

</body>

</html>

◆ **તમારી પ્રગતિ ચકાસો**

4) અમિત નામના સભ્ય દ્વારા ગ્રંથાલયમાંથી પુસ્તક ઇસ્યુ થયુ છે તે દર્શાવતુ SQL વિધાન લખો.

નોંધ : 1) તમારો જવાબ નીચે આપેલ જગ્યામાં લખો.

2) આ એકમને અંતે આપેલ જવાબ સાથે તમારો જવાબ સરખાવો.

....

....

....

....

....

....

3.5 પ્રોગ્રામિંગ ભાષાની પેઢીઓ (GENERATIONS OF PROGRAMMING LANGUAGES)

પ્રોગ્રામિંગ કોઈપણ ભાષા દ્વારા થઈ શકે જેવી કે C, CH, COBOL વગેરે, જે મોટાભાગે જરૂરિયાત પર આધાર રાખે છે. પરંતુ કૉમ્પ્યુટર સમાવિષ્ટ પ્રોગ્રામ પ્રારંભિક તબક્કે માત્ર બાયનરી સ્વરૂપમાં પ્રસ્તુત થવા જોઇએ. પ્રોગ્રામ કોઇપણ ભાષામાં લખાયેલ હોય તે કૉમ્પ્યુટરમાં કાર્યમાં લેવામાં આવે તે પહેલાં તેનાં બાયનરી પ્રદર્શનમાં અનુવાદિત કરવામાં આવે છે.

જેવી રીતે હાર્ડવેરનું પ્રોદ્યોગિકીને આધારે પેઢીઓમાં વર્ગીકરણ થયું છે તેથી જ રીતે કૉમ્પ્યુટર ભાષાનાં વર્ગીકરણ પણ યંત્ર સાથેનાં આદાન-પ્રદાનને આધારિત હોય છે. 1945ના વર્ષથી પાંચ વર્ગો અથવા પેઢીઓ પ્રોગ્રામીંગ ભાષાની છે જે વર્ષવાર નીચે મુજબ છે.

- પ્રથમ પેઢી 1945 યાંત્રિક ભાષા
- બીજી પેઢી 1950 એસેમ્બલી ભાષા
- ત્રીજી પેઢી 1960 ઊંચા સ્તરની ભાષા
- ચોથી પેઢી 1970 ખૂબ જ ઊંચા સ્તરની ભાષા
- પાંચમી પેઢી 1980 કુદરતી ભાષા

3.5.1 યાંત્રિક ભાષા (Machine Languages)

આ બાયનરી નંબરનાં સ્વરૂપમાં લખાયેલી સૂચનાઓની ક્રમબદ્ધતા છે. જેનાં 1 અને 0 ને જેને કૉમ્પ્યુટર સીધો પ્રતિભાવ આપે છે તેનો સમાવેશ થાય છે. યાંત્રિક ભાષા પ્રારંભમાં કોડ તરીકે ઓળખતી છતાં હવે કોડ શબ્દ વિશાળ રીતે કોઇપણ પ્રોગ્રામ લખાણને દર્શાવવા વપરાય છે.

સૂચના કોઇપણ યાંત્રિક ભાષામાં તૈયાર થાય તેને બે ભાગ હોય છે. પ્રથમ ભાગ કમાન્ડ અથવા સંચાલન. જે કૉમ્પ્યુટરને કયું કાર્ય કરવાનું છે તે કહે છે. સૂચનાનો બીજો ભાગ ઓપરેન્ડ છે. તે કૉમ્પ્યુટરને કયાં શોધવાનું છે અથવા રહેલા ડેટાને સંગ્રહ કરવાનું કહે છે. યાંત્રિક ભાષાને પ્રથમ પેઢીની ભાષા તરીકે કહેવામાં આવે છે.

3.5.2 એસેમ્બલી ભાષા (Assembly Languages)

જ્યારે નિશાનીઓ સંચાલન ભાગ તરીકે વાપરવામાં આવે, સૂચના કોડનાં સરનામા ભાગ અને અન્ય ભાગો, આ પ્રદર્શનને એસેમ્બલી ભાષા કહે છે. આને પેઢીની ભાષા કહેવામાં આવે છે. જ્યાં સુધી સમસ્યા વ્યક્તિગત સૂચના માટેની હોય, યાંત્રિક અને એસેમ્બલી ભાષાને નીચા સ્તરની ભાષા કહેવામાં આવે છે. દરેક યંત્રને પોતાની એસેમ્બલી ભાષા હોય છે કે જે પ્રક્રિયાનાં આંતરિક બંધારણ આધારિત હોય છે. યાંત્રિક અને એસેમ્બલી બંને ભાષાઓ યંત્ર આધારિત છે.

કૉમ્પ્યુટર પ્રોગ્રામ માત્ર યાંત્રિક ભાષામાં થાય છે, જો પ્રોગ્રામ કોઈ અન્ય ભાષામાં લખાયો હોય ત્યારે અનુવાદક કે મધ્યસ્થીની જરૂર પડે છે. એસેમ્બલર એવો ભાષા અનુવાદક છે જે તેનાં ઇનપુટને એસેમ્બલી ભાષા પ્રોગ્રામનાં સ્વરૂપમાં લે છે અને કેટલાક મેમરી સ્થાનમાં યાંત્રિક ભાષા કોડ પેદા કરે છે.

3.5.3 ઊંચા સ્તરની ભાષા (High Level Languages)

ઊંચા સ્તરની ભાષાને પ્રક્રિયા ભાષા પણ કહેવામાં આવે છે. પ્રોગ્રામીંગ ભાષાઓ જેવી કે C, CO-BOL, TORTRAN અને BASIC ઊંચા સ્તરની ભાષાઓ છે. જેવી રીતે યંત્ર અને એસેમ્બલી ભાષાનાં સર્જનનો ખર્ચ અને સમય વધ્યો, ઊંચા સ્તરની ભાષા વિકાસ પામી.

પ્રોગ્રામ ઊંચા સ્તરની ભાષામાં લખાયેલ હોય તે યંત્ર સમજે તેવા સ્વરૂપમાં અનુવાદિત કરાય છે અને તે સોફ્ટવેર દ્વારા થાય છે. જેને કમ્પાયલર કહે છે. જે સ્રોત કોડને ઇનપુટ તરીકે લે છે અને યાંત્રિક ભાષા કોડ કે જે યંત્રમાં હોય તેને આઉટપુટ તરીકે ઉત્પન્ન કરે છે. બીજા પ્રકારનો સોફ્ટવેર પણ અનુવાદ કરે છે. તેને ઇન્ટરપ્રિટર કહે છે.

3.5.4 ચોથી પેઢીની ભાષા (Forth Generation Languages)

ચોથી પેઢીની ભાષા 4GL તરીકે ઓળખાય છે. જે ઊંચા સ્તરની ભાષા છે. જેને કેટલીક સૂચનાઓની જરૂર પડે તે ત્રીજી પેઢી તેનું કાર્ય પરું કરે પછી તે પૂર્ણ કરે. આવી રીતે પ્રોગ્રામર ત્રીજી પેઢી કરતા 4GL માં ઝડપથી લખી શકે છે. ઘણી 4GL બિન-પ્રક્રિયા ભાષા છે. પ્રોગ્રામર પાસે પ્રોગ્રામ પ્રક્રિયાની માહિતી હોતી નથી પરંતુ તે ઇચ્છે છે તેનું વિવરણ થાય છે.

ઉદા. તરીકે તમારે ચોક્કસ બૂકની વિગતોની જરૂર છે. પ્રક્રિયા ભાષામાં પ્રોગ્રામર સૂચનાઓની શ્રેણી નીચેનાં તબક્કામાં લખશે.

સ્ટેપ-1 પત્રકમાંથી રેકોર્ડ પ્રાપ્ત કરો.

સ્ટેપ-2 જો તે રેકોર્ડ કૉમ્પ્યુટરનો પરિચય હોય, વિગતો જોવો

સ્ટેપ-3 જો તે રેકોર્ડ કૉમ્પ્યુટરનો પરિચય ન હોય, સ્ટેપમાં જોવો.

4GL માં પ્રોગ્રામર એક જ સૂચના લખે છે કે

પત્રકમાં કૉમ્પ્યુટરનાં પરિચય વિશેની માહિતી મેળવો.

ઘણી ચોથી પેઢીની ભાષા ફાઇલ અને ડેટાબેઝમાંથી માહિતી મેળવવામાં થાય છે અને તે માહિતીને દર્શાવે અથવા નકલ કરે. ચોથી પેઢીની ભાષામાં ભાષા સમસ્યા કે જે ભૂલનાં જવાબો અથવા ડેટાબેઝમાં ડેટાનાં પ્રશ્નોનો સમાવેશ થાય છે.

બંધારીત કચેરી ભાષા (SQL) એ ઉત્તમ ભાષા છે અને સાર્વત્રિક વપરાય છે. ઘણી અન્ય ભાષાઓની જેમ જેવી કે C, C++ પાસ્કલ અથવા જાવા, તમારો તમારા કૉમ્પ્યુટરમાં તેનો ઉપયોગ કરવાનો રસ્તો હોવો જોઇએ.

ઘણા ડેટાબેઝ અમલીકરણો SQLનો ઉપયોગ તેની ભાષા તરીકે ઉપયોગી કરે છે. જેમાંથી ડેટાબેઝ, લાઇક એડ, ડીબેટ, ડેટા વધારવાનું કાર્ય કરી શકે છે.

3.5.5 કુદરતી ભાષા (Natural Languages)

કુદરતી ભાષા બે પ્રકારની છે. પ્રથમ સામાન્ય માનવીય ભાષા જેવી કે અંગ્રેજી. બીજી પ્રોગ્રામિંગ ભાષા કે જે માનવીય ભાષાનો ઉપયોગ કરી લોકોને કૉમ્પ્યુટર સાથે આદાન-પ્રદાન કરાવે છે. કુદરતી ભાષા પ્રશ્ન અથવા કમાન્ડસ વધારે પ્રત્યાયન સ્વરૂપે મૂકે છે. કુદરતી ભાષા એ અભ્યાસનાં ક્ષેત્રનો ભાગ છે. જે આર્ટીફિસીયલ, ઇન્ટેલીજસ તરીકે ઓળખાય છે. આર્ટીફિસયલ ઇન્ટેલીજન્સ એ સંબંધિત પ્રોદ્યોગિકીનું સંગઠન છે કે જે યંત્રને માનવીય, ક્ષમતાઓ જેવી ક્ષમતાઓ વિકસાવવામાં સક્ષમ બનાવે છે. જેવી કે અભ્યાસ, રીઝનીંગ, પ્રત્યાયન, જોવું અને સાંભળવું.

◆ **તમારી પ્રગતિ ચકાસો**

(5) કેટલીક ત્રીજી પેઢીની ભાષાનાં નામ આપો.

નોંધ : (1) તમારો જવાબ નીચે આપેલ જગ્યામાં લખો.

(2) આ એકમને અંતે આપેલ જવાબ સાથે તમારો જવાબ તપાસો.

....

....

....

....

....

....

3.6 ઉપસંહાર (SUMMARY)

આ એકમમાં આપણે પ્રોગ્રામિંગ અને પ્રોગ્રામિંગ ભાષાઓ વિશે ભણ્યા. આ એકમનું મુખ્ય કેન્દ્રબિંદુ નીચે મુજબ છે.

- પ્રોગ્રામિંગ અને પ્રોગ્રામિંગ ભાષાનો ખ્યાલ
- પ્રોગ્રામિંગનાં તબક્કાઓ
- યંત્ર, એસેમ્બલી અને ઊંચા સ્તરની ભાષા
- સામાન્ય પ્રોગ્રામિંગ ભાષા અને ઉપયોગ

3.7 'તમારી પ્રગતિ ચકાસો'ના જવાબ (ANSWERS TO SELF CHECK EXERCISE)

(1) પ્રોગ્રામ

(2) અલગોરિધમમાં સ્પષ્ટ અને સરળ અર્થવાળા ચોક્કસ તબક્કાનો સમાવેશ થાય છે કે જે આપેલ પ્રારંભિક શરતો લઈ આવે છે, પ્રત્યાયન આઉટપુટને પેદા કરે છે અને ચોક્કસ સમયમાં નિર્ધારીત કરે છે. અલગોરીધમનું વિક્ટોરીયલ પ્રદર્શન ઉત્તમ નિશાની સ્રોતો ઉપયોગ કરે છે.

(3) પ્રોગ્રામિંગનાં 6 પગથિયાં આ મુજબ છે. 1) પ્રોગ્રામ વિવરણ 2) પ્રોગ્રામ આકાર, 3) પ્રોગ્રામ કોડ 4) પ્રોગ્રામ તપાસ, 5) પ્રોગ્રામ ડોક્યુમેન્ટેશન અને 6) પ્રોગ્રામની જાળવણી કોડીંગએ એક પ્રોગ્રામિંગનું પગથિયું છે જેમાં અસલી લખાણ અથવા કેચીંગનો પ્રોગ્રામિંગ ભાષામાં સૂચનાની વ્યવસ્થા તરીકે સમાવેશ થાય છે.

(4) SELECT BOOK _ ISSUED
FROM CIRCULATION _ DATA
WHERE BOKROWER _ NAME = "Amlt"

(5) કેટલીક ત્રીજી પેઢીની ભાષાઓ C, C++, BASIC અને FORTRAN

3.8 ચાવીરૂપ શબ્દો (KEY WORDS)

અમલીકરણ પ્રોગ્રામ : પ્રોગ્રામ જે ચોક્કસ સમસ્યાનાં નિરાકરણ માટે વિકસાવવામાં આવેલ હોય, સામાન્ય હેતુ કાર્ય માટે ઉપયોગી કાર્ય કરે અથવા મનોરંજન પુરું પાડે.

એસેમ્બલી ભાષા : નીચા સ્તરની પ્રોગ્રામિંગ ભાષા જે કૉમ્પ્યુટર વપરાશકર્તાને પ્રોગ્રામ ટૂંકાસ્વરૂપો લખયાને લખવા કહે છે અથવા નંબરને બદલે શબ્દ યાદ રાખવા સહેલાં છે.

કમ્પાઈલર : ભાષા અનુવાદક કે જે આખા પ્રોગ્રામને કૉમ્પ્યુટર ક્રિયા તે પહેલા ઊંચા સ્તરની ભાષામાંથી યંત્ર ભાષામાંથી યંત્ર ભાષામાં ફેરવે છે.

ઊંચા સ્તરની ભાષા : પ્રક્રિયા ભાષા તરીકે પણ જાણીતી છે તે અંગ્રેજી જેવી માનવભાષા જેવી લાગે છે.

ઇન્ટરપ્રિટર : ભાષા અનુવાદ જે દરેક પ્રક્રિયા વિધાનને યાંત્રિક ભાષામાં ફેરવે છે અને તરત જ નિયાન્વિત કરે છે.

યંત્ર ભાષા : બાયનરી પ્રકારની નીંચા સ્તરની ભાષા જેમા 1 અને 0 હોય છે જેથી કૉમ્પ્યુટર સીધું જ શરૂ થાય.

યંત્ર પ્રોગ્રામ : પ્રોગ્રામ જે કૉમ્પ્યુટરને જરૂરી સંચાલન કાર્યોમાં મદદ કરે અને અમલીકરણ પ્રોગ્રામને ચલાવવા સક્ષમ બનાવે.

3.9 સંદર્ભો અને આગળનું વાંચન (REFERENCE AND FURTHER READING)

Droomy, R.G. (2004). How to solve it by computer. New Delhi: Prentice Hall of India.

O'Leary, Timothy J. and O'Leary, Linda I. (2002), Computing essentials 2002-2003. International Edition. New York: McGraw-Hill Irwin.

Pratt, Terrence W. and Zelkowitz, Marvin V. (2003). Programming languages: design and implementation. 4th Edition. New Delhi: Pearson Education.

Schneider, G, Michael and Gersting, Judith L. (1998): An introduction to computer science. 2nd Edition. California: Brooks/Cole Publishing Company.

Williams, Brian K. and Sawyer, Stacey C. (2003). Using information technology: a practical introduction to computers and communications. 5th Edition. New Delhi: Tata McGraw-Hill Publishing.