



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-વિસ્તૃત

```
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)

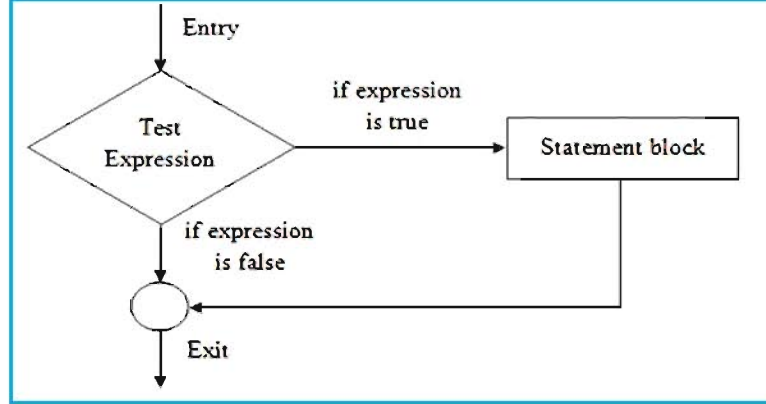
નીચેની વાક્યરચના દ્વારા સરળ if વિધાનનો ઉપયોગ કરી શકાય :

```
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)

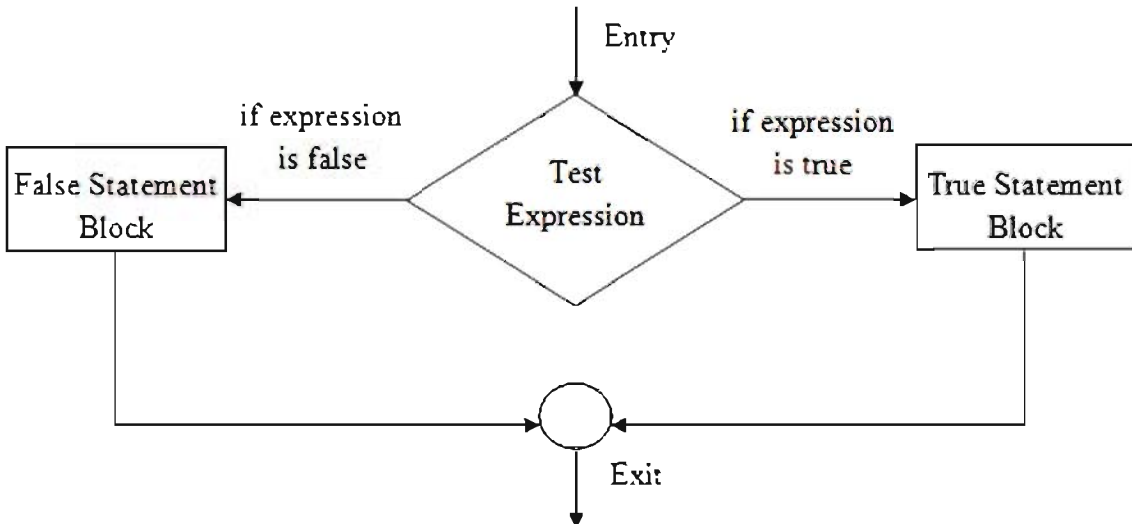
if...else વિધાનની વાક્યરચના નીચે આપેલ છે :

```

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

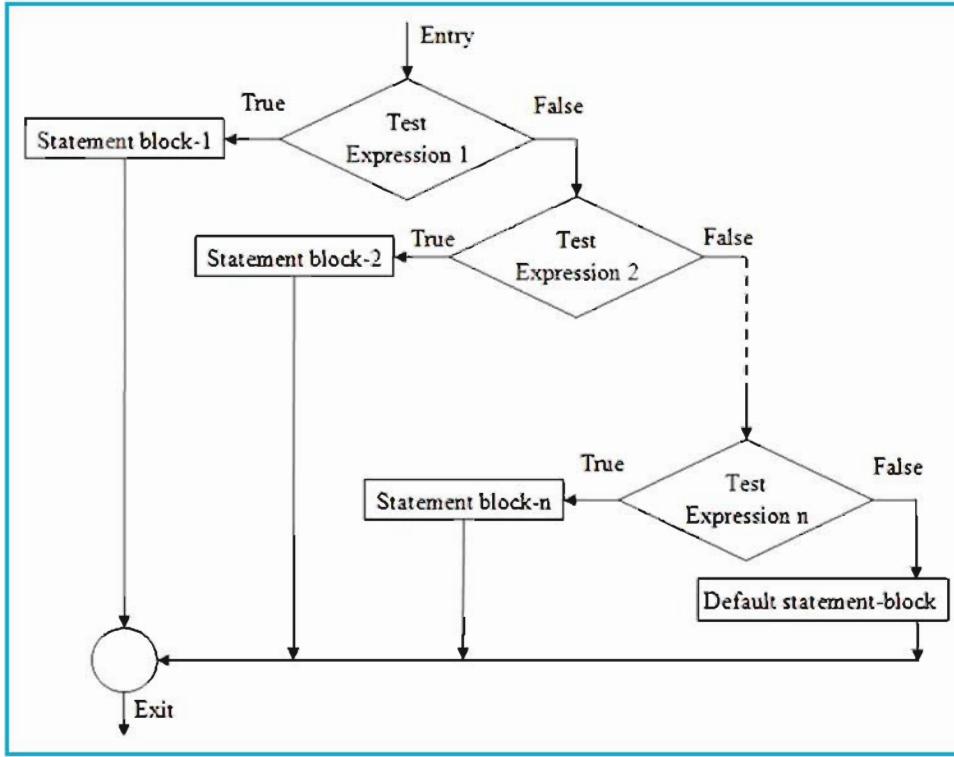
સમજૂતી

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 સંકળાયેલાં 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)

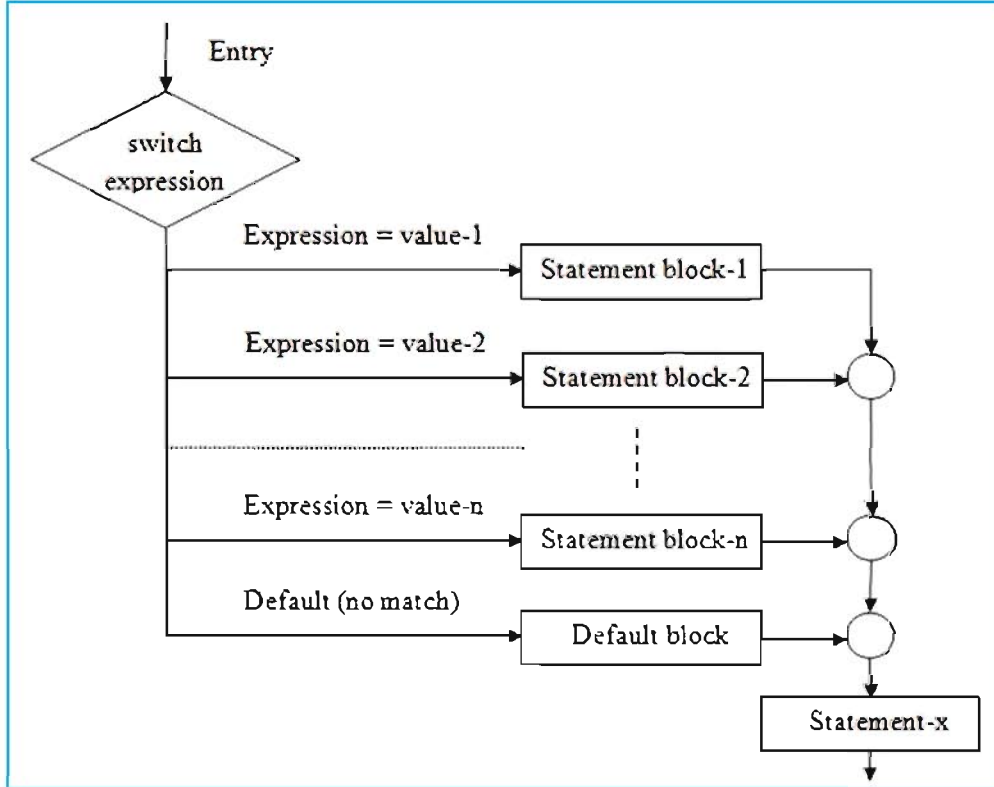
```

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.

