UNIT-3: Decision Making statements

3.1 if statements

3.1.1 simple if statements
3.1.2 if…else statements
3.1.3 if…else if….else statements
3.1.4 Nested if statements

3.2 switch..case statements

3.2.1 Use of break and default
3.2.2 Difference between switch and if statements

Decision Making Statements

In C programming, decision-making statements are used to control the flow of execution based on certain conditions.The primary decision-making statements in C are:

1. if statment
2. switch statement

These decision-making statements help in creating logic and controlling the program's flow based on different conditions.

3.1 if statements

In C programming, the "if" statement is a conditional statement that allows you to execute a block of code only if a specified condition is true. There are several types of "if" statements that can be used in C.

3.1.1 simple if statements

The simple "if" statement checks a single condition and executes a block of code if the condition is true. The syntax is as follows.If the test expression is true, then the statement block is executed; otherwise, the statement block is skipped, and execution will jump to statement-x.

syntax

```
if (test expression)
{
        Statement-block;
}
Statement-x;
```

Flowchart of simple if statement

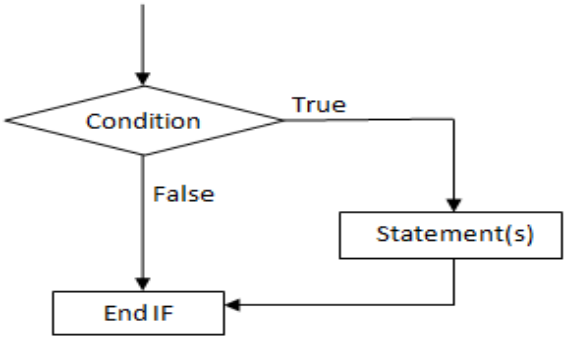


fig: Flowchart for if statement

Example 3.1:Check if the given number is a positive number.

```
#include <conio.h>
#include <stdio.h>
void main()
{
        int n;
        clrscr();
        printf("\n enter no:");
        scanf("%d",&n);
        if(n>0)
                printf("\n %d is positive number",n);
        getch();
}
```

Output:
Enter no: 5
5 is positive number

3.1.2 if…else statements

If the test expression is true, then the true statement block is executed; otherwise, the false statement block is executed.

syntax:

```
if (test expression)
{
        True-Statement-block;
}
else
{
        False-Statement-block;
}
Statement-x;
```

Flowchart of if…else statements

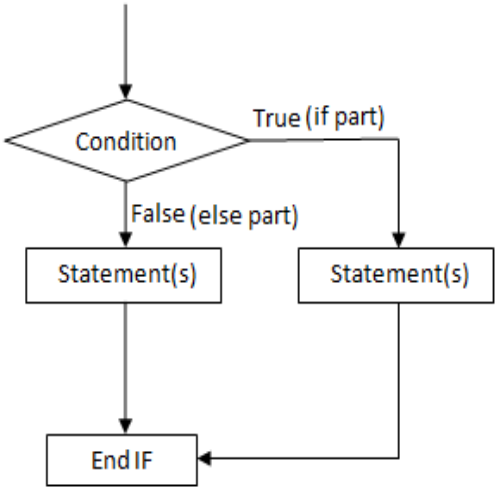


fig: Flowchart for if ... else statement

Example 3.2 Check if the given number is an even or odd number.

```
#include <conio.h>
#include <stdio.h>
void main()
{
        int n;
        clrscr();
        printf("\n enter no:");
        scanf("%d",&n);
        if(n%2==0)
                printf("\n %d is even number",n);
        else
                printf("\n %d is odd number",n);
        getch();
}
```

Output:

```
Enter no:5
5 is odd number
```

3.1.3 if…else if….else statements (THE ELSE IF LADDER)

This construct is known as an "else if" ladder. The tests are evaluated from top to bottom. As soon as a true test is found, the statement associated with it is executed, and control transfers to statement-x. When all the tests become false, the final "else" containing the default statement will be executed.

syntax:

```
if (test1)
        Statement-1;
elseif (test2)
        Statement-2;
elseif (test3)
        Statement-3;
else if (testn)
         Statement-n;
else
        Default statement;
Statement-x;
```

Flowchart of if…else if…else statments

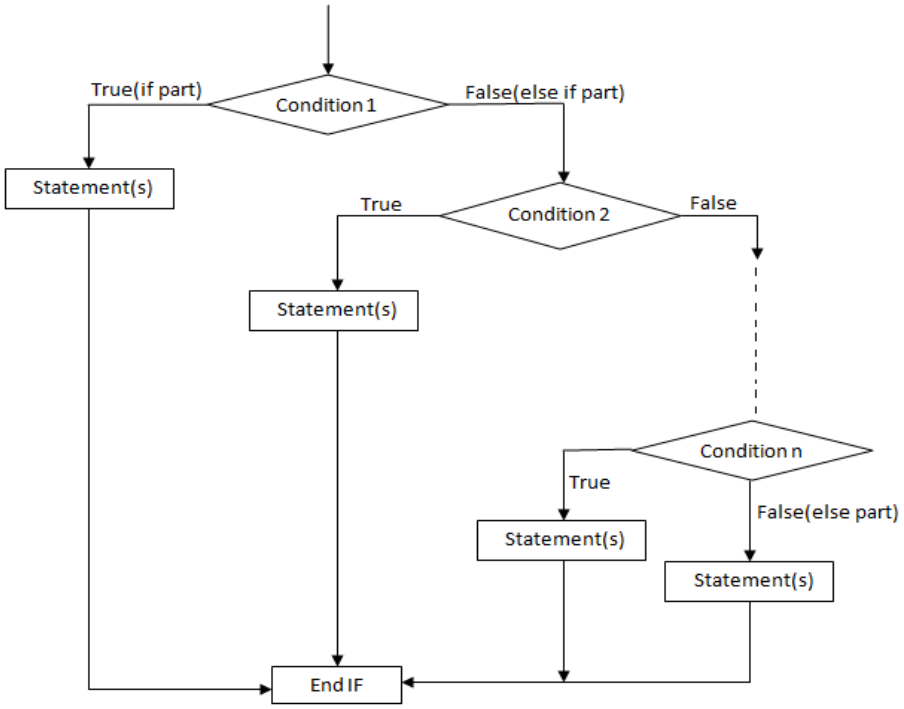


fig: Flowchart for if ... else if ... else statement

Example 3.3 Display marksheet using "else if" ladder.

```
#include <conio.h>
#include <stdio.h>

void main()
{
        int rn,m1,m2,m3,m4,m5,t,p;
        char name[20];
        clrscr();
        printf("\n Enter Name Rn M1 M2 M3 M4 M5 \n");
        scanf("%s %d %d %d %d %d %d",name,&rn,&m1,&m2,&m3,&m4,&m5);
        t=m1+m2+m3+m4+m5;
        p=t/5;
        printf("\n Name is:%s",name);
        printf("\n Roll no:%d",rn);
        printf("\n Marks1:%d",m1);
        printf("\n Marks2:%d",m2);
        printf("\n Marks3:%d",m3);
        printf("\n Marks4:%d",m4);
        printf("\n Marks5:%d",m5);
        printf("\n Total is:%d",t);
        printf("\n Percentage is:%d",p);
        if (p>70)
                printf("\n Distinction");
        else if(p>60)
                printf("\n First class");
        else if(p>50)
```

```
                printf("\n Second class");
        else if(p>40)
                printf("\n Pass class");
        else
                printf("\n Fail");
        getch();
 }
```

Output:

```
Enter name,rn,m1,m2,m3,m4,m5
Sima 1 80 80 80 80 80

Name is: Sima
Roll no:1
Marks1:80
Marks2:80
Marks3:80
Marks4:80
Marks5:80
Total is: 400
Percentage is:80
Distinction
```

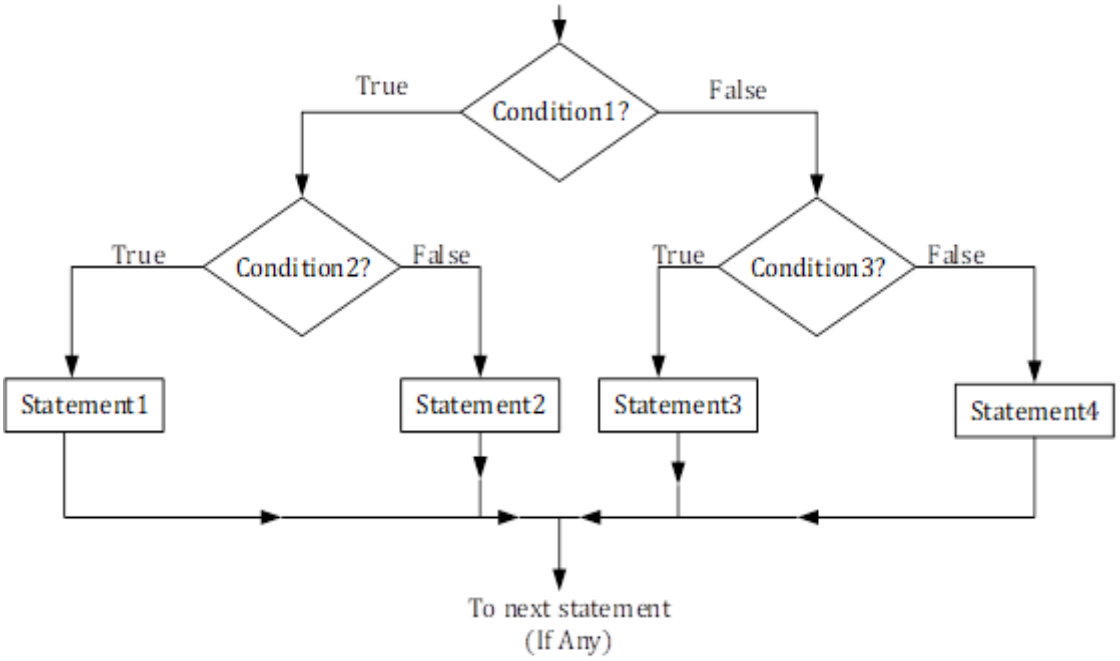
3.1.4 Nested if statements

If an "if" statement is contained within another "if" statement, it is called a nested if.
If the test expression1 is true, then it checks test expression2. If test expression2 is true, statement-1 is executed; otherwise, statement-2 is executed.
If test expression1 is false, then it checks test expression3. If the test expression3 is true, statement-3 is executed; otherwise, statement-4 is executed. Control then transfers to statement-x.

```
if (test expression1)
{
        if(test expression2)
        {
                Statement-1;
        }
else
        {
                 Statement-2;
        }
}
else
{
        if(test expression3)
        {
                Statement-3;
        }
else
        {
                 Statement-4;
        }
}
```

Statement-x;

Flowchart of Nested if statements



Example 3.4 Find the largest number among three numbers.

```
#include <conio.h>
#include <stdio.h>

void main()
{
        int a,b,c;
        clrscr();
        printf("\n enter a,b,c:");
        scanf("%d %d %d",&a,&b,&c);
        if(a>b)
            if(a>c)
                printf("\n %d is largest number",a);
            else
                printf("\n %d is largest number",c);
        else
            if(b>c)
                printf("\n %d is largest number",b);
            else
                printf("\n %d is largest number",c);

        getch();
}
```

Output:
Enter a,b,c 12 50 5

50 is largest number

3.2 Switch..case statements

3.2.1 Use of break and default

The expression is an integer expression or character. Value1, value2, and so on are constants or constant expressions known as case labels. Each of these values should be unique within a switch statement. Block1, block2, and so on are lists of statements and may contain zero or more statements. There is no need to put braces around these blocks. Case labels end with a colon (:).

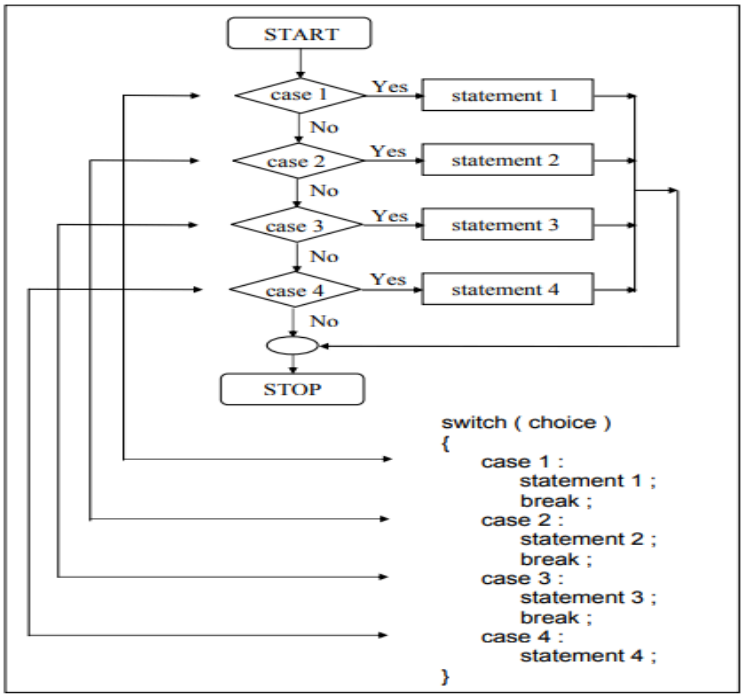
When the switch is executed, the values of the expression are compared against the values value1, value2, and so on. If a case is found whose value matches the expression, the block of statements following the case is executed. The break statement at the end of each block indicates the end of a particular case and causes an exit from the switch statement, transferring control to statement-x.

The default case is optional. When present, it will be executed if the expression's value does not match any of the case values. If not present, no action takes place if all matches fail, and control goes to statement-x.

```
syntax
switch (expression)
{
        case value1:
                Block1;
                break;
        case value2:
                Block2;
                break;
                .
                .
        default:
                default-block;
                break;

        Statement-x;
}
```

Flowchart of Switch…case statements



Example 3.5: Switch case example using an integer type choice.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int d;
    clrscr();
    printf("\n enter day in number:");
    scanf("%d",&d);
     switch(d)
     {
        case 1:
                printf("\n monday");
                break;
        case 2:
                printf("\n tuesday");
                break;
        case 3:
                printf("\n wednesday");
                break;
        case 4:
                printf("\n thursday");
                break;
        case 5:
                printf("\n friday");
                break;
        case 6:
                printf("\n saturday");
                break;
        case 7:
                printf("\n sunday");
```

```
                break;
        default:
         printf("\n invalid day number");
        break;
        }
         getch();
    }
```

Output:
enter day in number:1
monday

Example 3.6: Switch case example using a character type choice

```
#include <stdio.h>
#include <conio.h>
void main()
 {
 char c;
 clrscr();
 printf("\n enter alphabet:");
 scanf("%c",&c);
 switch(c)
 {
        case 'A':
                printf("\n alphabet is A");
                break;
        case 'B':
                printf("\n alphabet is B");
                break;
        case 'C':
                printf("\n alphabet is C");
                break;
        case 'D':
                printf("\n alphabet is D");
                break;
        default:
         printf("\n invalid input");
         break;
        }

 getch();
 }
```

Output:
Enter alphabet: A
Alphabet is A

3.2.2 Difference between switch and if statements.

IF	SWITCH
Expression inside if statement decides whether to execute the statements inside if block or under else block.	The expression inside the switch statement determines which case to execute.
An if-else statement uses multiple expressions for multiple choices.	The switch statement uses a single expression to handle multiple choices.
If statement checks for equality as well as for logical expression.	Switch checks only for equality.
The if statement evaluates integer, character,pointer or floating point type or boolean type.	It evaluates only character or an integer data type.
If the expression inside the if statement evaluates to false, the statement inside the else block will be executed.	If the expression inside the switch statement evaluates to false, the default statement will be executed.
Editing if-else statements can be challenging.	Editing switch statements is straightforward.

Programming Exercises

1. Write a C program to input two numbers and find the maximum number from the two.

```
#include <conio.h>
#include <stdio.h>
void main()
{
  int a,b;
  clrscr();
  printf("\n enter no a and b:");
  scanf("%d %d",&a,&b);
  if(a>b)
          printf("\n %d is maximum number",a);
  else
          printf("\n %d is maximum number",b);
  getch();
}
```

2. Write a C program to input a year (Y) and determine whether it is a leap year.

```
#include <conio.h>
#include <stdio.h>
void main()
{
int y;
clrscr();
printf("\n enter year:");
scanf("%d",&y);
```

```
if(y%4==0)
  printf("\n %d is leap year",y);
else
  printf("\n %d is not leap year",y);
getch();
}
```

3. Write a C program to check if a given number is positive, negative, or zero.

```
#include <conio.h>
#include <stdio.h>
void main()
{
  int n;
  clrscr();
  printf("\n enter no:");
  scanf("%d",&n);
  if(n>0)
        printf("\n %d is positive number",n);
  else if(n<0)
        printf("\n %d is negative number",n);
  else
        printf("\n number is zero");
  getch();
}
```

4. Write a C program to perform addition, subtraction, multiplication, and division using a switch case.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a,b,c;
  clrscr();
  printf("\n enter a and b");
  scanf("%d %d",&a,&b);
  printf("\n Enter Choice:");
  scanf("%d",&c);
  switch(c)
  {
        case 1:
              printf("\n Addition is %d",a+b);
              break;
        case 2:
              printf("\n Subtraction is %d",a-b);
              break;
        case 3:
              printf("\n Multiplication is %d",a*b);
              break;
        case 4:
```

```
                printf("\n Division is %d",a/b);
                break;
        Default:
                printf("\n invalid input");
                break;
}
getch();
}
```