

**GOVERNMENT COLLEGE FOR WOMEN(A)**

**KUMBAKONAM**

**DEPARTMENT OF COMPUTER SCIENCE**

**"PROGRAMMING IN C"**

**TOPIC : DECISION MAKING AND BRANCHING STATEMENTS**

**STAFF NAME : S.SUNDARESWARI**

**DATE & TIME : 01/10/2020 & 9.30am to 11.30am.**

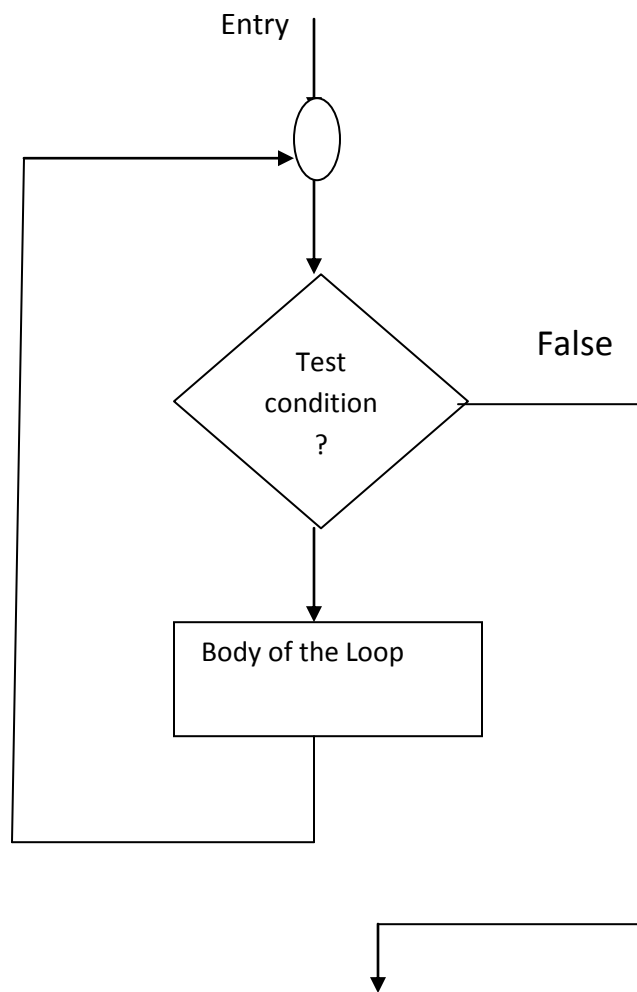
## DECISION MAKING AND LOOPING :

In looping ,a sequence of statements are executed until some conditions for termination of the loop are satisfied.

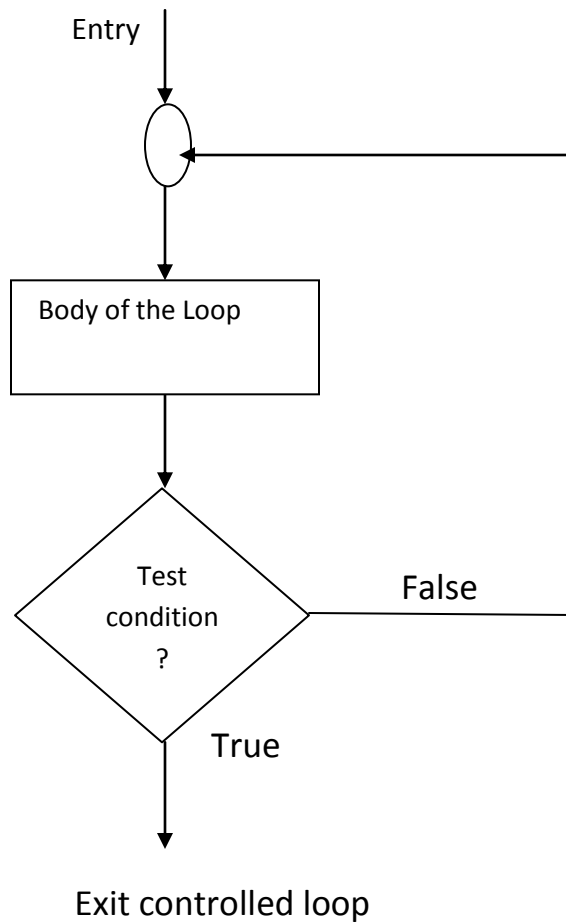
A **program loop** therefore consists of two segments, one known as the body of the loop and the other known as the control statements.

A control structure is classified into two types :

- 1) **Entry-controlled loop (For loop , While loop)**
- 2) **Exit-controlled loop (do...while)**



Entry controlled loop



**C programming has three types of loops:**

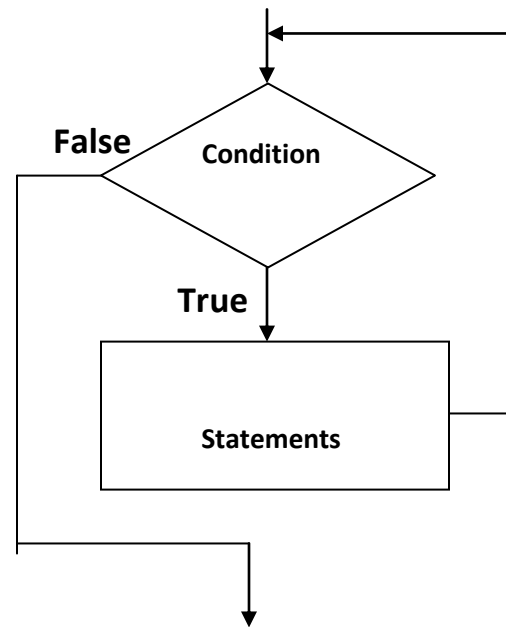
- 1) The `while` loop**
- 2) The `do...while` loop**
- 3) The `for` loop**

## 1) The while statement :

The while loop is used when a statement is to be executed repeatedly until a given condition is satisfied. The loop is terminated when the expression yields false value.

The general form is

```
while (condition)
{
    Statements;
}
```



The while is entry controlled loop statement. The test condition is evaluated and if the condition is true, the body of the loop is executed. After execution of the body, the test condition is once again evaluated and if it is true, the body is executed once again. This process of repeated execution of the body continues until the test condition finally becomes false and the control is transferred out of the loop. On exit, the program continues with the statement immediately after the body of the loop.

## Program Example : Reverse Number

```
#include<stdio.h>

#include<conio.h>

void main( )

{

    Int m,n,k=1;

    printf ("Enter the number\n");

    scanf ("%d", &n);

    printf ("The Reverse number is \n");

    while ( k != 0)

    {

        m = n % 10;

        k = n / 10;

        n = k;

        printf ("%d", m);

    }

    getch();

}
```

Input : Enter a number

**3456**

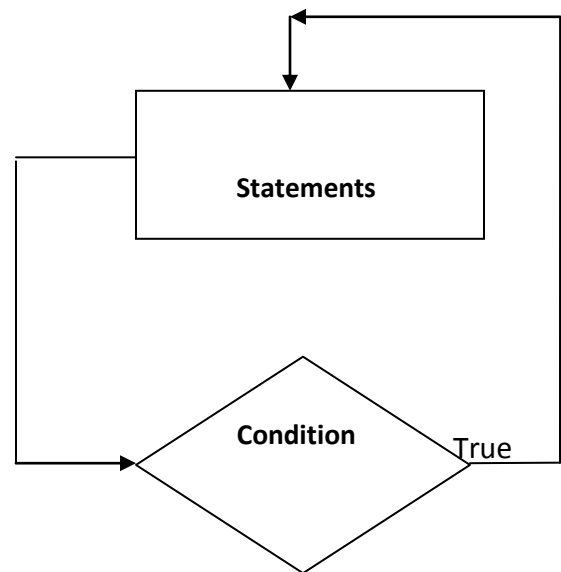
Output : 6543

## 2) The do...while statement :

This is very similar to the while loop except that the test occurs at the end of the loop body. This ensures that the loop body is run at least once.

The general format is :

```
do
{
    Statements;
} while (condition);
```



In do..while statement ,the program proceeds to evaluate the body of the loop first. At the end of the loop, the test condition in the while statement is evaluated. If the condition is true, the program continues to evaluate the body of the loop once again. This process is continues as long as the condition is true. When the condition becomes false, the loop will be terminated and the control goes to the statement that appears immediately after the while statement.

## PROGRAM EXAMPLE :

```
#include <stdio.h>
#include<conio.h>
void main()
{
    double n, sum = 0;
    do
    {
        printf ("Enter a number :\n");
        scanf (" %lf " , &n);
        sum = sum + n;
    } while(n != 0.0);
    printf (" Sum = %2lf " ,sum);
    getch();
}
```

### Output :

Enter a number : 1.5

Enter a number : 2.4

Enter a number : -3.4

Enter a number : 4.2

Enter a number : 0

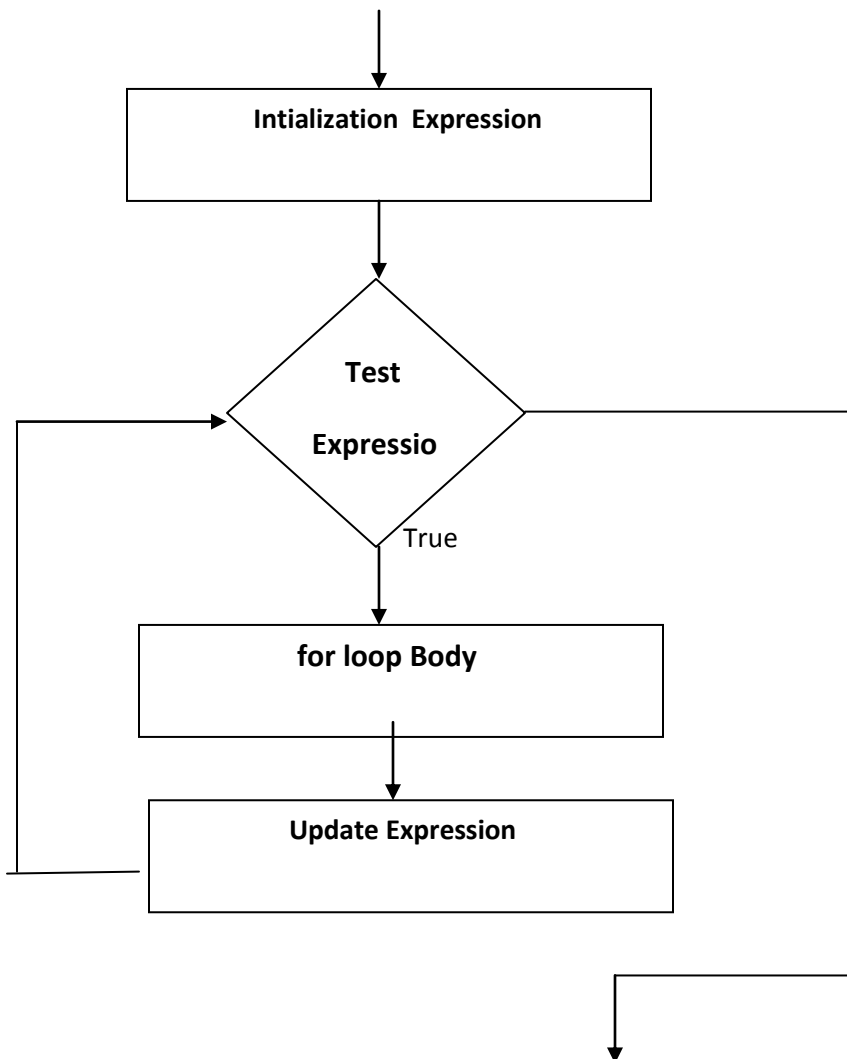
**Sum = 4.70**

### 3) The For loop :

The for loop works well where the number of iterations of the loop is known before the loop is entered.

The general form of the for loop is

```
for ( initialization; test-condition ; increment )  
{  
    Body of the loop  
}
```





## How for loop works?

- The initialization statement is executed only once.
- Then, the test expression is evaluated. If the test expression is evaluated to false, the for loop is terminated.
- However, if the test expression is evaluated to true, statements inside the body of for loop are executed.
- Again the test expression is evaluated.

This process goes on until the test expression is false. When the test expression is false, the loop terminates.

### PROGRAM EXAMPLE : FACTORIAL NUMBER

```
#include<stdio.h>
#include<conio.h>
void main ( )
{
    int n, f =1, i=1;
    printf ("Enter the number\n");
    scanf ("%d", &n);
    for (i=1; i <= n; i++)
    {
        f = f * i;
    }
}
```

```
printf( "The factorial number is %d ", f);  
}
```

**Output :**

**Enter the number : 3**

**The factorial number is 6**