Aug-18

UNIT-III

SELECTION- STATEMENTS

* Sometimes it is desirable to alter the Sequential flow of control to provide for a choice of action. This Requires a logical test based on a test Expression to be carried out at Some particular point with in the program.

  The action will then be carried out depending on the outcome of logical test. This is called "Condition Execution".

* Conditional Execution, in which one group of statement is selected from several available groups, is known as "SELECTION".

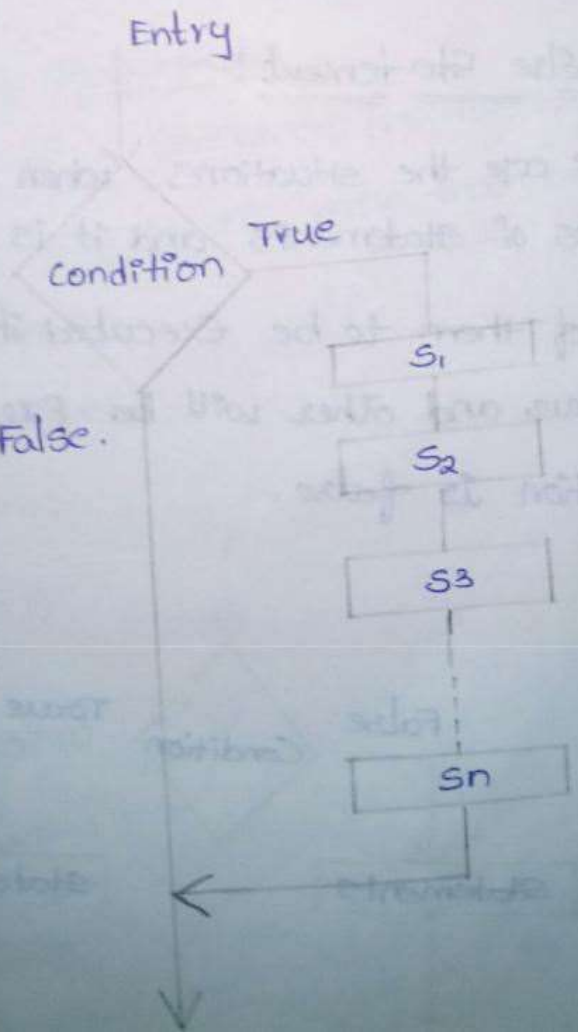
1) Simple if statement
2) if else statement
3) Nested if (statement) - else statement.
4) else if ladder.
5) Switch statement.
6) Ternary statement.

* Simple-if statement:-

The if statement is used to specified Condition Executions, programm statements (or) a group of statements.

The general format is if (Condition)

```
{
  statement 1;
  statement 2;
  |
  |
}
```



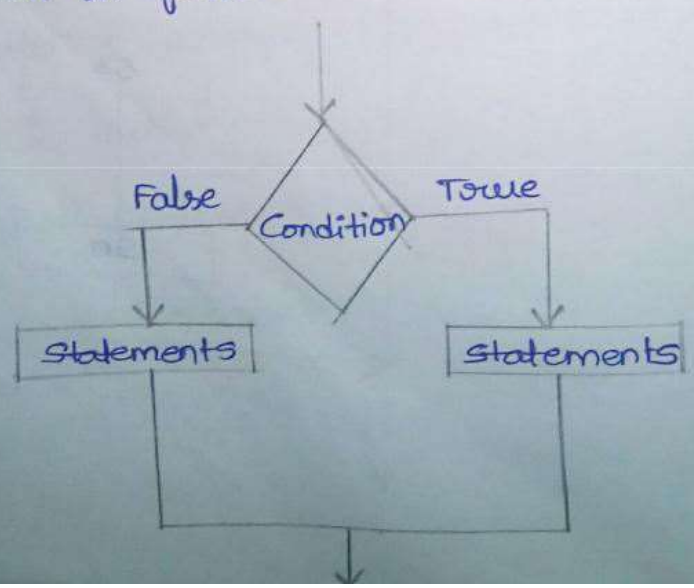
Example:-

```
main( )
{
int n;
printf("Enter n value");
scanf("%d", &n);
if (n==0)
printf("you entered zero");
}
```

Aug-19 *

If-Else Statement:-

There are the situations when there are two groups of statements and it is decided. One of them to be executed if some condition is true and other will be executed if the condition is false.



/* Biggest of two Numbers */

Eg:-
```
main( )
{
int a,b;
printf ("Enter a,b values");
scanf ("%d %d", &a, &b);
if (a>b)
printf ("a is big");
else
printf ("b is big or Equal");
}
```

* **Nested if else statement:-**

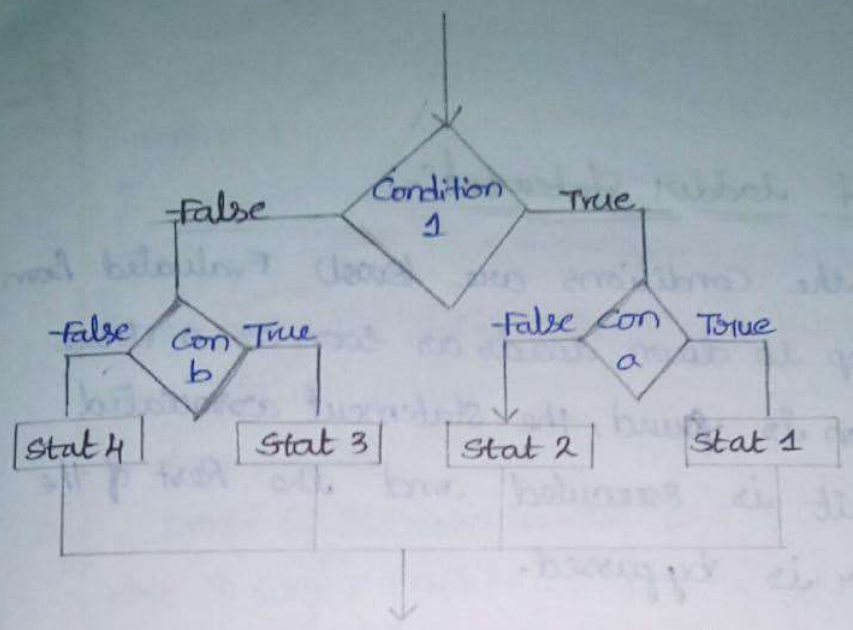
when a series of a (~~distance~~) Resistance are Involved, they may to use more than one if-Else statement. (Nested)

In Nested form

The general form:-

```
if (condition 1)
{
  if ( conditiona)
  {
    statement 1;
  }
  else
  {
```

```
        statement 2;
    }
}
else
{
  if (condition b)
  {
  statement 3;
  }
  else
  {
  statement 4;
  }
}
```



Ex:- /* Biggest of three numbers */

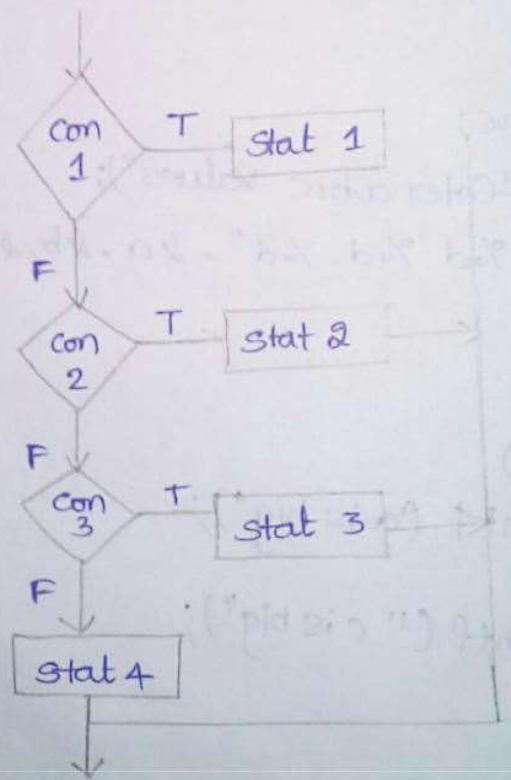
```
main( )
{
 int a,b,c;
 printf("Enter a,b,c values");
 scanf("%d %d %d", &a, &b, &c);
 if (a>b)
 {
  if (a>c)
      printf ("a is big");
  Else
      printf (" c is big");
 }
 Else
 {
  if (b>c)
      printf ("b is big");
  Else printf ("c is big");
 }
```

9.

* **else-if ladder statement:-**

Here the conditions are (Excel) Evaluated from the top to down loads as soon as a true condition is found, the statement associated with it is executed and the Rest of the ladder is bypassed.

The last else is handles (default case.




```
if (condition 1)
    statement 1;
else if (Condition 2)
    Statement 2;
else if (Condition 3)
    statement 3;
else
    statement 4;
```

Eg:-

```
main( )
{
 int age ;
 printf ("Enter age Value");
scanf ( "%d", & age);
if (age >= 80)
   printf (" sty fund is 1000 ");
else if (age > = 70)
   printf ("styfund is 800");
else if (age > = 60)
  printf ("Styfund is 600");
else
   printf (" No styfund ");
}.
```

* Switch Statement :-

Switch statement works on same way as if-else-if but it is more eligant.

The switch statement is the special multi way decision maker that test whether ~~(decis) (Resist)~~ * An Expression Matches one of may have number of Constant values and branch according to the Conditions.

```
switch (condition)
{
case value 1 : statement 1 ; break;
case value 2 : statement 2 ; break;
case value 3 : statement 3; break;
        |
        |
        |
case value n : statement n; break;
default : statement , break,
}.
```

Aug 20

Ex:-

```
main ( )
{
int n;
printf ("Enter number");
scanf ("%d", &n);
switch(n)
{
case 1 : printf ("SUNDAY"); break;
case 2 : printf ("MONDAY"); break;
    |
default : printf("No day"); break;
}
}
```

* if (n==1)
p("SUNDAY");
else if (n==2)
p("MON");

* A switch statement tests the value of given variable expression against a list of values.

when a Match is found a block of statements associated with the case is Executed.

In the above Example, the Expression is an Integer expression and the values are 1,2,3,4,5,6,7.
The break statement is End of Each drop.

* TERNARY STATEMENT:-

'C' provides Conditional Evaluation (operator) operator is called Ternary Statement.

in the form of ? :

The general format is (condition) ? Exp1 : Exp;
The ? operator relates the condition that preceeds it. if it is true, it Returns Exp 1, else it returns Exp 2.

Ex:-

```
main( )
{
int a,b;
printf ("Enter two numbers");
scanf ("%d %d", &a, &b);
c = (a>b) ? a;b;
printf (" max is %d", c);
}.
```

ITERATIVE STATEMENTS (REPEATATION)

* These statements are allow a set of Instructions to be performed until a certain condition is Reached.

'c' provides three different types of loops.

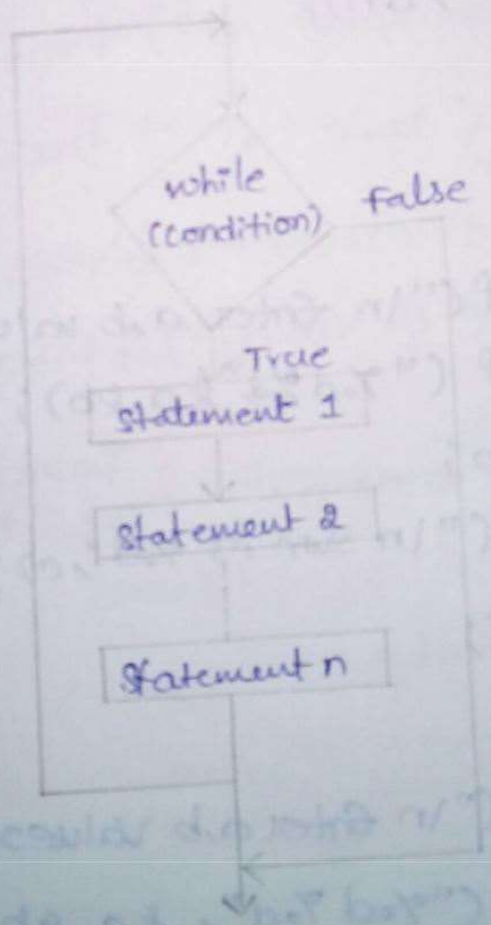
namely ; i while loop
ii do-while loop
iii for loop

i, WHILE LOOP:-

The while loop in the 'c' starts with while Key word, followed by a parameters rolling condition, and has a set of statements which Constitute the body of the loop.

The general format is;

```
while (condition)
{
 statement 1;
 statement 2;
      :
 statement n;
}
```



Switch Statement

*

```
#include <stdio.h>
main( )
{
int a,b,c,n;
float f;
printf("\n1:add 2.sub 3.mul 4.div");
printf("\nEnterNumber ");
scanf("%d", &n);
switch(n)
{
case 1 ;
        printf("\n Enter a,b values");
        scanf ("%d %d", &a,&b);
        c=a+b;
        printf("\n Sum is %d",c);
        break;
case 2 ;
        printf("\n Enter a,b values");
        scanf("%d %d", &a,&b);
        c=a-b;
        printf("\n sub is %d",c);
        break;
```

```
case 3 :
        print f ("\n Enter a,b values");
        scanf ("%d %d", &a, &b);
        c = a*b;
        printf ("\n Mul is %d",c);
        break;

case 4 :
        printf ("\n Enter a,b values");
        scanf ("%d %d", &a, &b);
&       (c=a/b) f=(float)a/b;
        printf ("\n div is %f",f);
        break;
default : printf("\n there is no operation");
}       break;
}.
```

Aug 26

* As soon As Execution reaches while loop, the Specified Condition is tested if it is form to be true, it will Enter into the body of the loop. Once it Reaches the closing ~~Rays~~ trace of the body. automatically loop back to the top and test the Condition forcely now And if it is true re-Enter the body and so on. till the Controlling Condition of the loop becomes false.

Eg:-

```
while (condition)
{
  ==
}.
```

* Factorial *

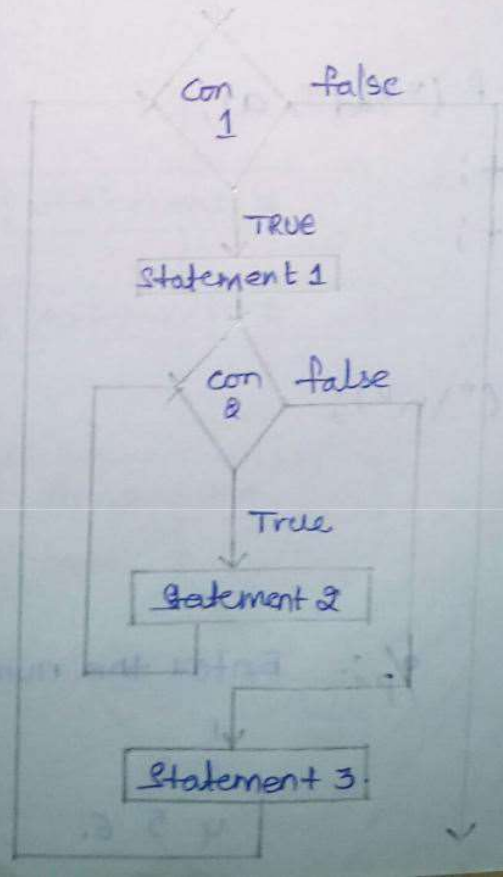
```
#include <stdio.h>
# include <conio.h>
main ( )
{
int n,i=1,f=1;
clrscr( );
printf(" \n Enter n value");
scanf ("%d", &n);
```

```
while (i<=n)
{
f= f*i;
i= i+1;
}
printf ("\n factorial is %d", f);
getch();
}.
```

Nested Loop's

b) when a while loop is Executed, instead of there is one or more Internal loops. Those loops are called "Nested loops".

The general format is



* /* floyd's triangle */

```
#include <stdio.h>
main( )
{
int n,i,k,a=1;
printf ("Enter the number");
scanf ("%d", &n);
i=1;
while (i<=n)
{
 k=1;
  while (k<=i)
  {
   printf ("%d", a);
    a++;
    k++;
  }
 printf ("\n");
 i++;
}
}
```

o/p :- Enter the number 3

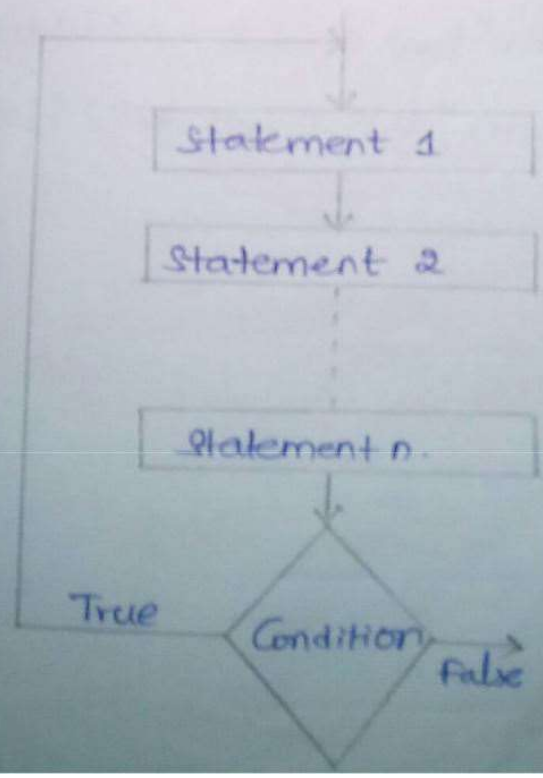
```
1
2 3
4 5 6.
```

* Do-while loop:-

The Do while loop performes the test at the bottom Rather than at the top. The Do while loop starts with key word DO, ~~followed~~ -followed by the body of the loop.

The general format is

```
do {
    Statement 1;
    Statement 2;
    :
    Statement n;
} while (Condition);
```



*Eg:-

```
main( )
{
int n,i=1,f=1;
printf("Enter n value");
scanf("%d", &n);
do {
f=f*i;
i=i+1;
}while(i<=n)
printf("factorial is %d", f);
}.
```

Aug 29

Add Digits & Multiplication of a Number

Eg:-

```
main( )
{
 int n, Sum = 0, mul = 1, r;
printf ("enter Number");
Scanf ("%d", &n);
while (n!=0)
{
  r = n % 10;
  n = n/10;
  Sum = Sum + r;
  mul = mul * r;
}
printf ("\n Sum is %d", sum);
printf ("\n Mul is %d", mul);
}.
```

Arm Strong Number

```
*   main( )
{
 int n,r, arm=0,temp;
 printf("Enter number");
 scanf("%d", &n);
 temp = n;
 while (n! =0).
 {
   r = n%10;
   n= n/10;
   arm = arm + (r*r*r);
 }
 if (temp = =arm)
   printf("In Arm Strong");
 Else
   printf("In Not arm strong");
}
```

Aug 31

* (c) For loop:-

This is used when the statements are to be Executed more than once.

This is the most quitely used iteration loop.

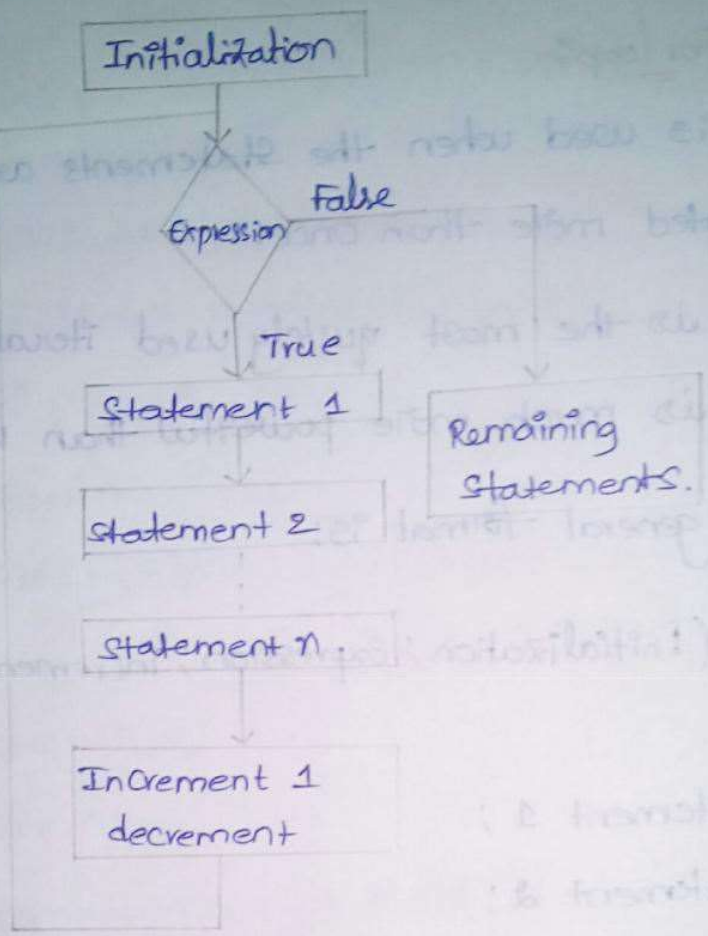
This is much more powerful than while loop.

The general format is:-

```
for (Initialization ; Expression ; increment/Decrement)
{
  Statement 1 ;
  Statement 2 ;
      :
  Statement n ;
}
Remaining statements;
```

Eg:-

```
main( )
{
int n,i,f=1;
printf ("Enter Number");
scanf ("%d", &n);
for (i=1; i<=n; i++)
{
 f = f*i;
}
printf ("\n fact is %D", f);
}
```



Eg:-

```
main( )
{
int n,i=1;
printf ("Enter number");
Scanf("%d", &n);
for (i<=10; i<=n; i++)
{
printf ("\n %d * %d", n, i, n*i);
}
}
```

```
main()
{
int n,i,k;
clrscr();
printf("Enter n value");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
for (k=1;k<=i;k++)
{
printf("%d",k);
}
printf("\n");
}
}.
```

Sep-01

JUMP STATEMENTS

* (I) break
  (II) Continue
  (III) go to

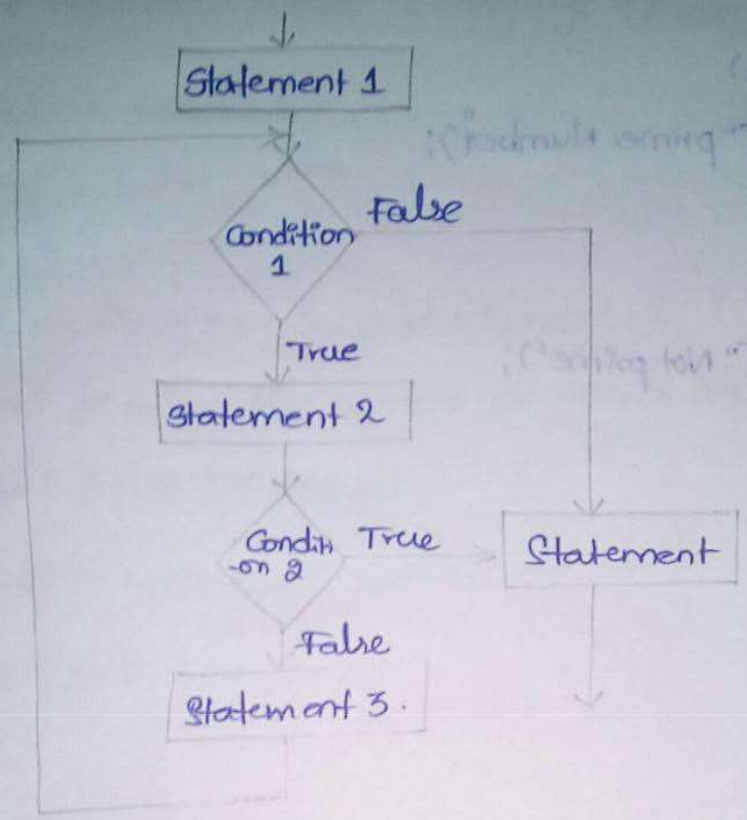
* (I) break:-

In 'c' programming the break statement can break the inner most control structure. we already have used the break statement in switch statement and also use inside a while loop, a for loop and do-while loop.

The general format is

```
Statements 1;
while (Condition)
{
 Statement 2;
 if (Condition 2)
  break;
 Statement 3;
}
 Statement 4;
```

Eg:



Eg: For loop ⇒ prime Number

```
#include <stdio.h>
main( )
{
int n, i=1, f=0;
printf ("Enter the number");
scanf ("%d", &n);
for (i=1 ; i≤n ; i++)
{
 if (n%i==0)
{
 f=f+1;
}
```

```
}
if (f==2)
printf ("prime Number");
else
{
printf (" Not prime");
}
}
```

```
/* Fibonacci Series c language */

#include <stdio.h>
main( )
{
int n, first=0, second=1, next, i;
printf ("Enter the number of terms \n");
scanf ("%d", &n);
printf ("Fibonacci series are", n);
for (i=0; i<n; i++)
{
 if (i<=1)
   next=i;
   else
 {
  next=first+second;
  first= Second;
  Second=next;
 }
 printf ("%d", next);
 }
}
```

output

Enter the number of terms 5

fibonacci series are 0 1 1 2 3

- 8

- 0 1 1 2 3 5 8 13

- 12

- 0 1 1 2 3 5 8 13 21 34 55 89

2. pascal

```
#include<stdio.h>
long fact (int);
main( )
{
int i,n,c;
print f("Enter the number of rows");
Scan f ("%d", &n);
for ( i=0; i<n; i++)
{
for (c=0; c<=(n-i-2); c++)
Printf (" ");
print f ("%d", fact (i)/fact (c) * fact(i-c)); for (c=0,c<=i; c++)
print f ("\n")
}
}
long fact (int n)
{
int c;
long result = 1;
for (c=1; c<=n; c++)
```

```
result = result * c;
return result;
}
```

output

Sep-20

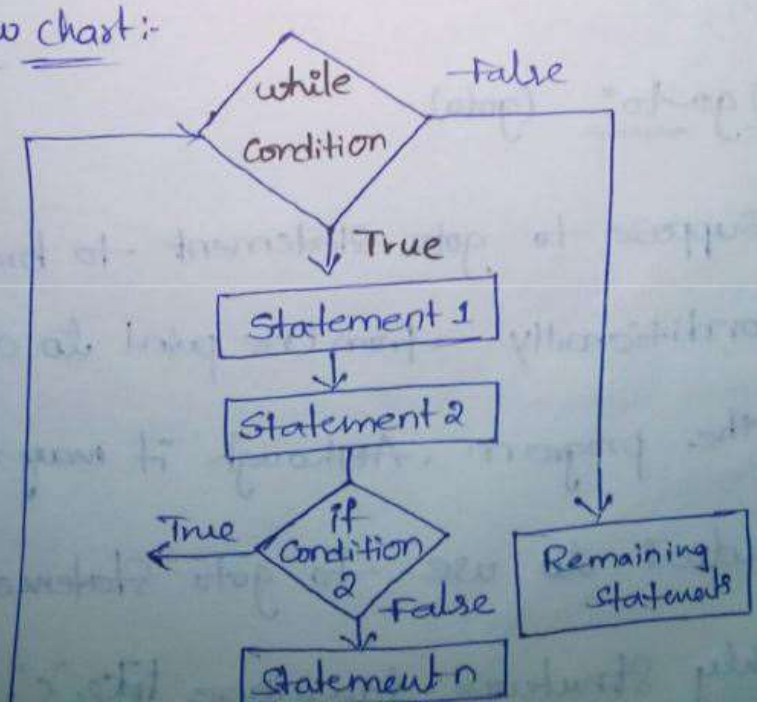
* (ii) Continue :-

The Continue statement, whenever Executed the Rest of current Iteration to be scipped and causes the next iteration to begin.

The general format is,

```
while (Condition)
{
    Statement 1;
    Statement 2;
    if (condition 2)
        Continue;
    Statement n;
}
```

Flow chart :-



* (go to:)

Example:

①
```
#include <stdio.h>
main( )
{
int i;
while (i<=10)
{
printf (" i value is %d ", i);
if (i==5)
Continue;
i++;
}
}
```

②
```
main( )
{
int i=0;
while (i<=10)
{
i++;
if (i==5)
Continue;
printf (" i value is %d ", i);
}
}
```

(iii) go to:- (goto).

c Suppose to goto statement to branch unconditionally from one point to another point in the program. Although it may not be Essential to use to goto statement in a highly structure language like 'c'.

The goto Requires a variable in order to identify its place where the branch is to be made.

The general format :-

```
        statement 1;
        goto memo;
        statement 2;
        statement 3;
           :
        statement n;
memo:   statement n+1;
        statement n+2;
        statement n+3;
```

1. $x = 1 + 2 + 3 + 4 + --- + n$

```
#include <stdio.h>
#include <conio.h>
main( )
{
int i=1, n, x=0;
printf ("Enter the Number");
scanf ("%d", &n);
while (i<=n)
 {
  x=x+1;
   i++;
 }
printf ("x is %d", x)
}
```

2. $x = 1^2 + 2^2 + 3^2 + .. + n^2$

```
#include <stdio.h>
main( )
{
int i=1, n, x=0;
printf ("Enter the number");
```

```
scanf("%d", &n)
while (i<=n)
 {
 x=x+i*i)
    i++;
 }
printf ("x is %d", x);
}
```

3. $x = 1/1^2 + 1/2^2 + 1/3^2 + \cdots 1/n^2$

```
#include <stdio.h>
main( )
{
int i=1, n, x=0;
                 → float x=0.0;
printf ("enter the number");
scanf ("%d", &n);
while (i<=n)
{
x = x+1/(i*i);
   i++;
}
printf ("x is %d", x)
                  %f
}.
```