

*Eg:-

```
mainc )
{
    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 29Add Digits & Multiplication of a NumberEg:-

```
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 quickly used iteration loop.

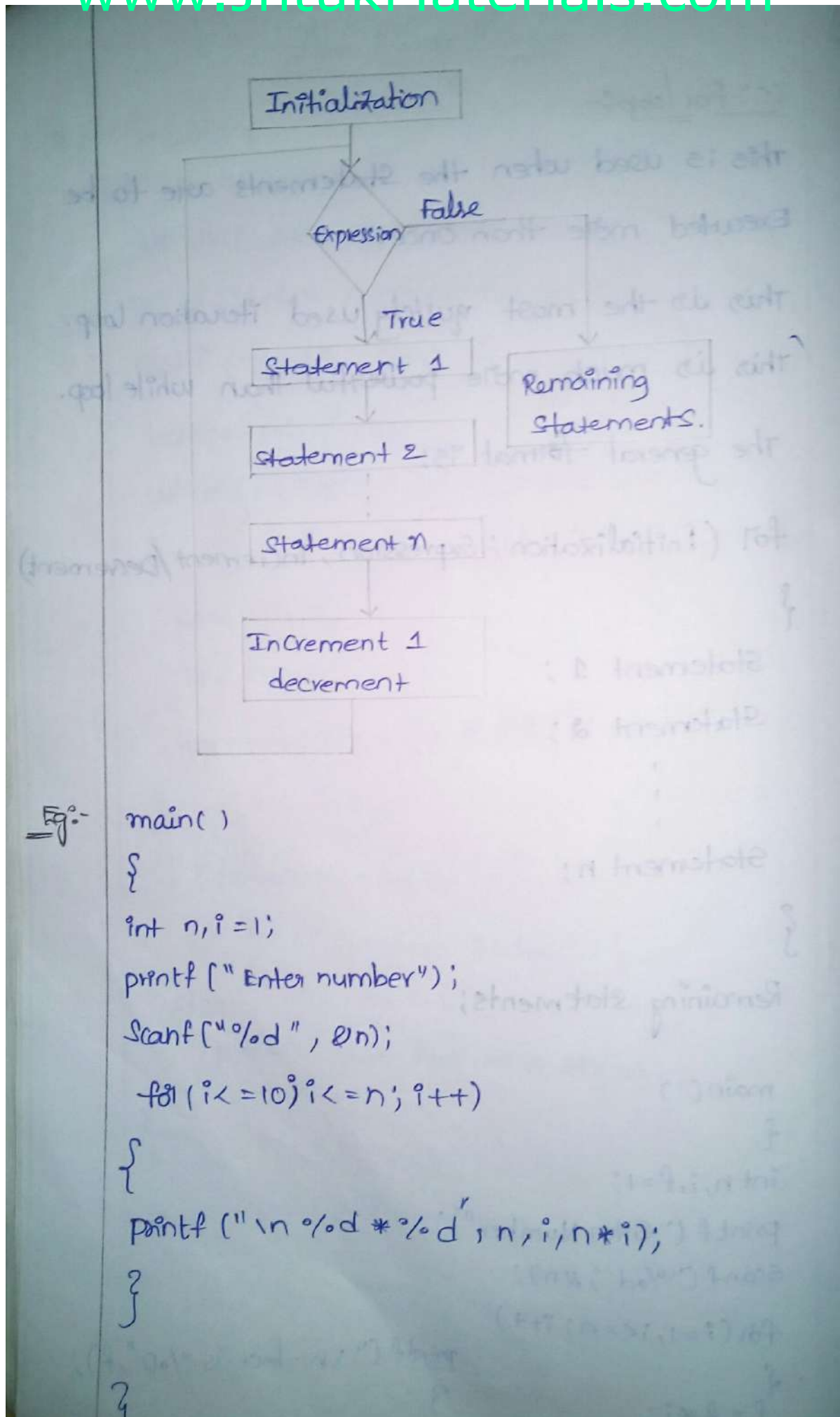
This is much more powerful than while loop.

The general format is:-

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

Ex:

```
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);
    }
}
```



```
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:-

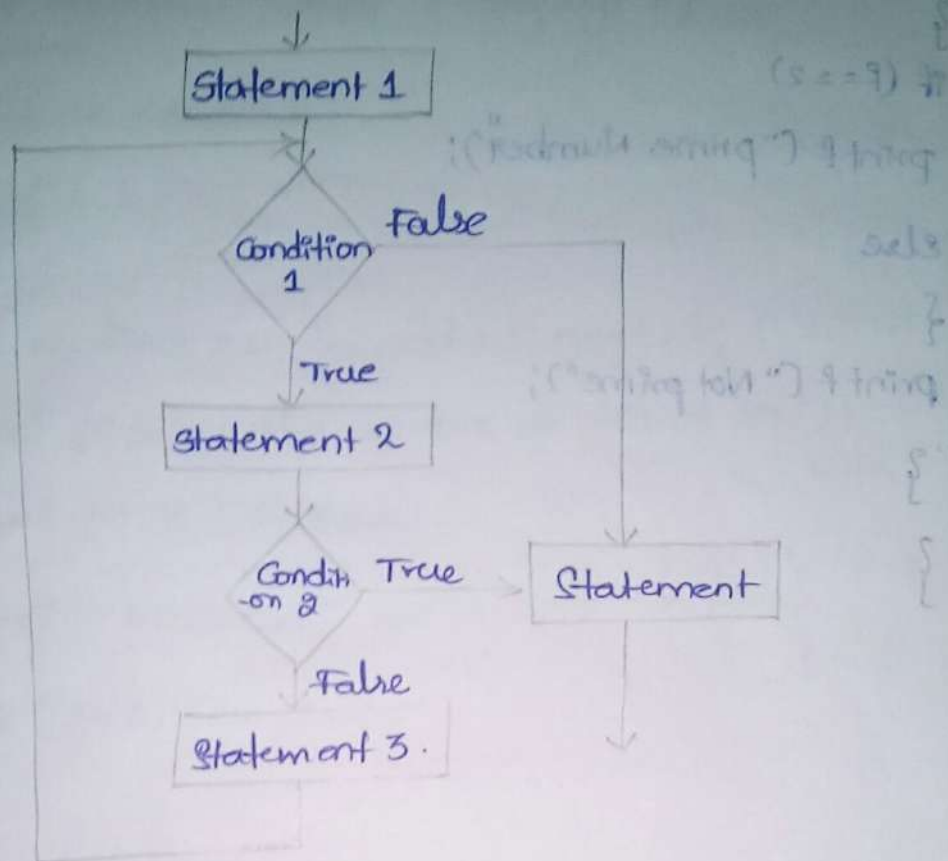
In 'c' programming the break Statement Can break the inner most Control Structure. we already have used the break statement in Switch statement and also use inside a while loop, a for loop and do-while loop.

The general format is

```

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

Eg:



Eg:

for loop $\Rightarrow$ 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 (p == 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 5

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;

printf ("Enter the number of rows");

scanf ("%d", &n);

for (i=0; i<n; i++)

{

for (c=0; c<=(n-i-2); c++)

printf (" ");

for (c=0; c<=i; c++)

printf ("%d", fact(i)/fact(c) * fact(i-c));

printf ("\n");

}

}

long fact (int n)

{

int c;

long result = 1;

for (c=1; c<=n; c++)

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

output

Sep-20

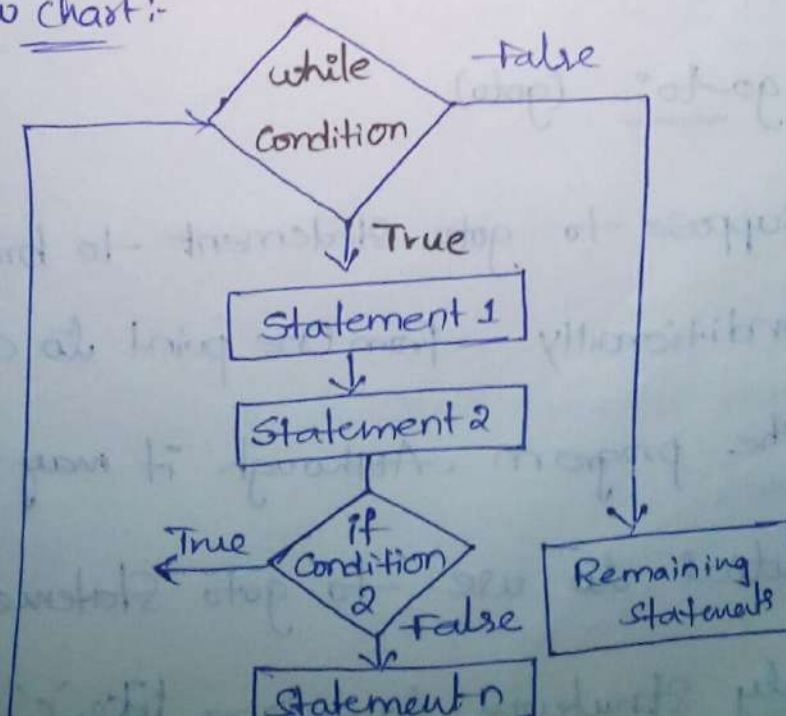
*

(ii) Continue:-

The Continue statement, whenever Executed the Rest of current Iteration to be skipped 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. $x = 1 + 2 + 3 + 4 + \dots + n$

```
#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 + i;
```

```
    i++;
```

```
}
```

```
printf ("x is %d", x)
```

```
}
```

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

```
#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. $x = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2}$

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

```
main( )
```

```
{
```

```
    int i=1, n, x=0;
     $\rightarrow$  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)
```

```
};
```