Control Statement.

In c-programming the program is executed from top to bottom. Each instruction are executed only once. Sometimes, we might need to break the sequence of the program and jump from one instruction to another as per the requirements. Control statements are those statement which control the flow of execution of program. Mainly control statements are of followings categories:-

1) Branching

a) Conditional statements
   i) If statement
   ii) If..... else statement.
   iii) If..... else If statement.
   iv) Nested if..... else statement.

b) Switch case statements

2) Looping.

a) while loop
b) Do..... while loop
c) for loop.

3) Jumping statement

a) break;
b) continue;
c) Goto;
d) Exit();

1) Branching:

It is also known as selective control structure in which selection is made on the basis of condition. When the condition is true the program will go towards certain statements & otherway when the condition is false. With the help of branching the programmer can also give user the choices to execute the selective instruction.

Branching can be classified into two types:

a) conditional statement
b) switch case

a) Conditional or control structure:

It specifies the order in which various instructions or statements are to be executed in a program. There are four types of control structure.

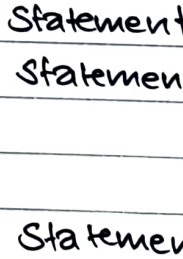
i) Sequence:

It is a control structure in which statement are executed sequentially

Syntax:

```
Statement 1;
Statement 2;
:
:
Statement n;
```

Flowchart



ii) Selective/decision :-

It is used to control the flow of program based upon condition. Conditional statement are mainly used for decision making various decision making various decisions control statement.

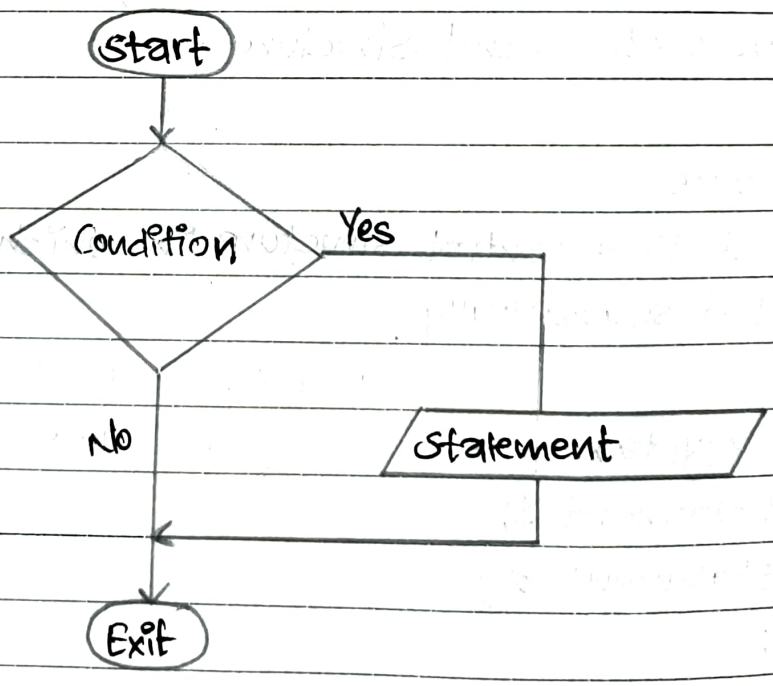
a) if statement :-

It is the simplest format conditional statement. If the given condition is true then it will execute the given statement otherwise the statement outside this structure is executed

Syntax:

```
if (condition)
{
  statement /expression;
}
```

Flowchart:



Date :
Page :

WAP to enter user's age & display you can cast vote. If the user is >= 18

```
#include <stdio.h>
#include <conio.h>
  void main()
 {
   int a;
   clrscr();
   printf("\n Enter your age : ");
   scanf ("%d", &a);
   if (a >= 18)
   {
     printf("\n You can vote");
   }
   getch();
 }
```

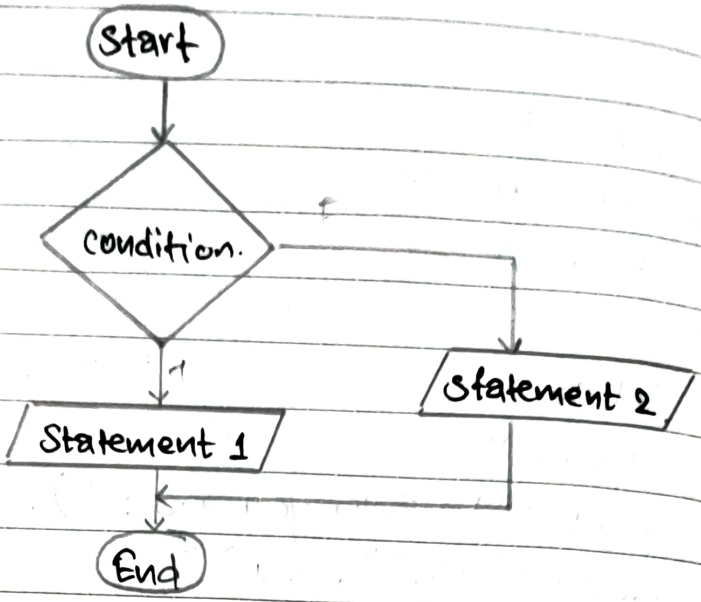
b) if else statement:

In another form at selective control structure which can handle both true and false states of a given condition.

Syntax:

```
if (condition)
{
  statement 1;
}
else
 {
   statement 2;
 }
```

Flowchart :



WAP to check whether the number enter is odd or even.

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
    int a,x;
    clrscr();
    printf("\n Enter any integer:");
    scanf("%d",&a);
     x=a%2;
     if(x==0)
     {
       printf("\n %d is even number",a);
     }
       else
         {
           printf("\n%d is odd number",a);
         }
       getch();
}
```

Date : .....
Page: .....

WAP to check whether the num is +ve or -ve

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
    int a;
    clrscr();
    printf("\n Enter any number : ");
    scanf ("%d" ,&a);
    if (a>0);
    {
      printf ("\n %d is +ve no ", a);
    }
      else
      {
        printf ("\n %d is -ve no", a);
      }
      getch();
  }
```

C) if else ..... if statement

It is also known as multiple conditional statement or multipath conditional statement or if else ladder. It is used when we have to check two or more condition

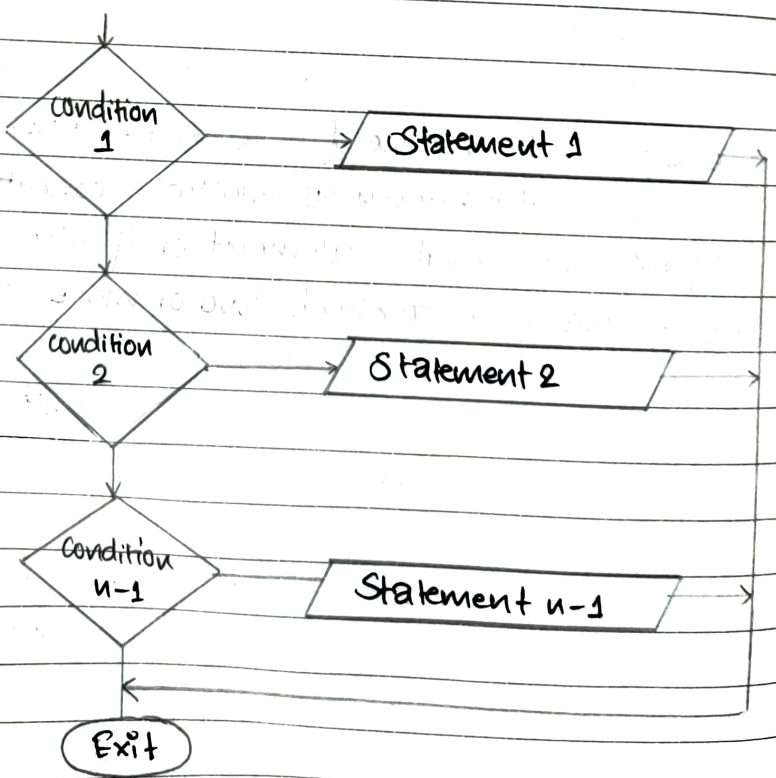
→

Page:

Syntax:

```
if (condition 1)
 {
  statement 1;
 }
else if (condition 2)
 {
  statement 2;
 }
else if (condition n-1)
  {
   statement n-1;
  }
else
  {
   statement n;
  }
```

Flowchart:



WAP to find the greatest among three numbers.

```
#include <stdio.h>
#include <conio.h>
 void main()
 {
    int a, b, c;
    clrscr();
    printf("\n Enter any three number: ");
    scanf("%d %d %d", &a, &b, &c);
    if (a>b && a>c)
    {
      printf("\n %d is the greatest ", a);
    }
    else if (b>a && b>c)
    {
      printf("\n %d is the greatest ", b);
    }
    else
    {
      printf("\n %d is the greatest ", c);
    }
    getch();
 }
```

/* Smallest no */

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a, b, c;
  clrscr();
  printf("\n enter three no : ");
  scanf("%d %d %d", &a, &b, &c);
  if (a<b && a<c)
  {
    printf("\n %d is the smallest ", a);
  }
  else if (b<a && b<c)
  {
    printf("\n %d is the smallest ", b);
  }
  else if (c<a && c<b)
  {
    printf("\n %d is the smallest ", c);
  }
  else
  {
    printf("\n %d Numbers are equal");
  }
  getch();
}
```

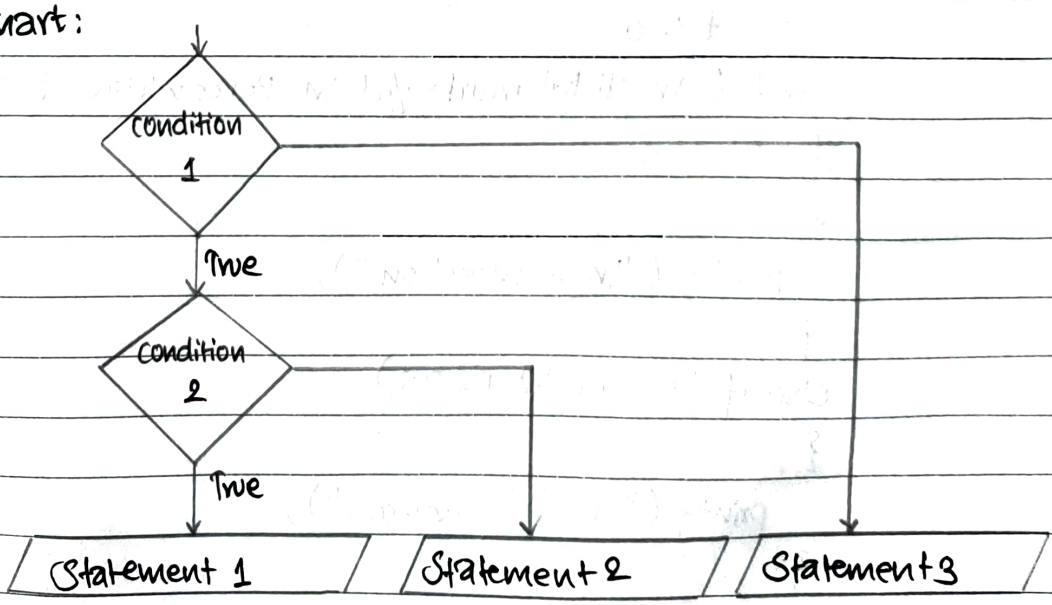
d) Nested if else statement:

An entire if else statement can be written within the body of if part or else part of another if else statement is called nested if else statement.

Syntax:

```
if (condition)
{
   if (condition 2)
     {
      statement 1;
     }
   else
     {
      statement 2;
     }
else
 {
   statement 3;
 }
```

flowchart:



WAP to enter marks of five different subjects & calculate total marks & percentage & display division on the basis of following criteria.

Percentage	Division
P >= 75	Distinction
P >= 60 & P < 75	First division
P >= 45 & P < 60	Second division
P >= 35 & P < 45	Third division
P > 35	Fail

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int eco, comp, eng, math, acc, t;
    float P;
    printf("\n Enter marks of economics, english, math,
            accounts & computer : ");
    scanf("%d %d %d %d %d", &eco, &comp, &eng,
          &math, &acc &com);
      t = eco + comp + eng + math + acc;
      P = t/50;
    printf("\n Total marks %d \n Percentage is %f", t, P);
    if (P>=75)
    {
      printf("\n Distinction");
    }
    else if (P>=60 && P<75)
    {
      printf("\n 1st division");
    }
```

```
else if (p>=45 && p<60)
 {
    printf("\n 2nd division");
 }
 else if (p>= 35 && p<45)
 {
    printf("\n 3rd division");
 }
 else
 {
    printf ("\n You have failed");
 }
 getch();
}
```

2) Switch case statement:

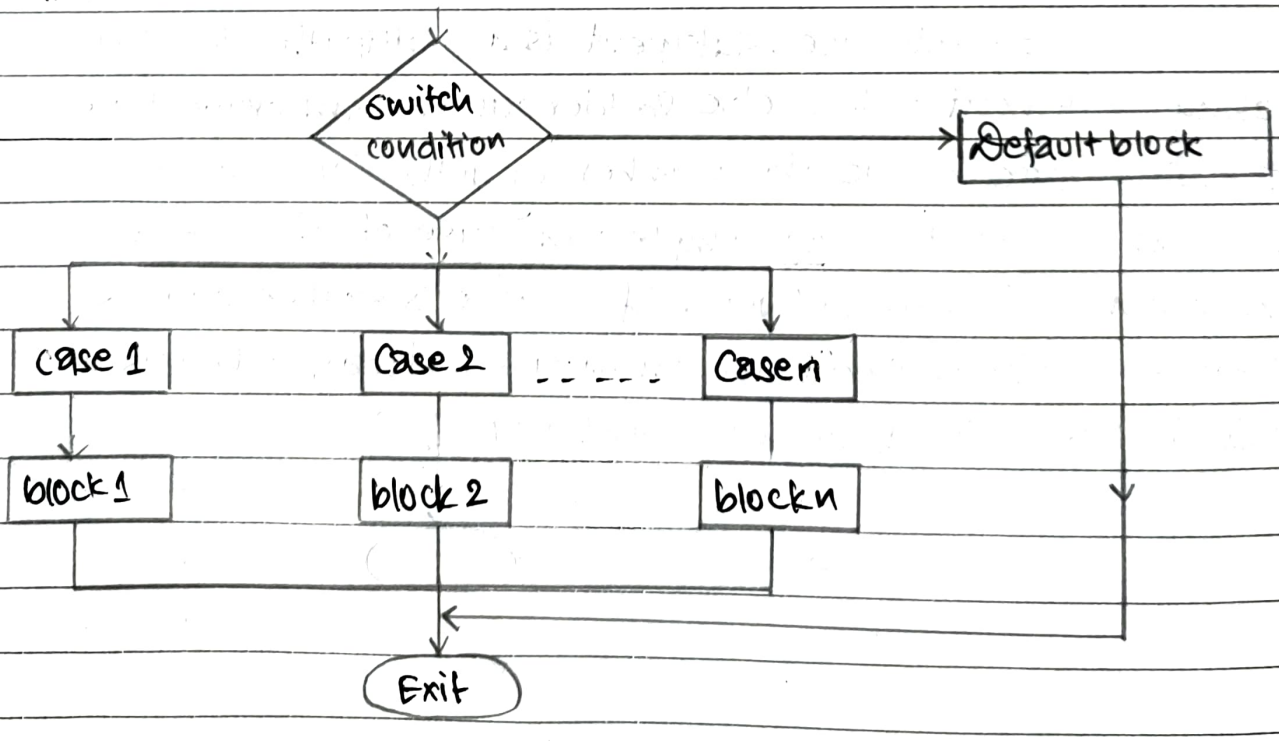
Switch case statement is a multipath decision making statement. If.....else ladder can perform some task as switch case but as the number of authority increases selection process becomes complex in case of if.....else The main difference between if.....else & switch case is that if.....else makes selection in a serial way whereas in switch case it is done in parallel way.

→

Syntax:

```
Switch (expression)
{
 case constant 1 ,
  Statement 1;
    break;
   :
   :
  case constant n-1.
   statement n-1;
    break;
  default;
   statement n;
}
```

Flowchart:



```
#include <stdio.h>
#include <conio.h>
void main()
{
int ch;
clrscr();
printf("\n Enter choice :");
scanf("%d", &ch);
switch(ch)
{
case 1: printf("\n First);
        break;
case 2: printf("\n Second");
        break;
case 3: printf("\n Third");
        break;
default: printf("\n wrong choice");
}
getch();
}
```

WAP to enter two number and ask the user to display sum, diff & product according to their choice using switch case statement.

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
    int a,b,s,ch;
     clrscr();
     printf ("\n Enter two number : ");
     scanf ("%d %d", &a, &b);
     printf ("\n Press 1 for add");
     printf ("\n Press 2 for subtract");
     printf ("\n Press 3 for multiplication");
     printf ("\n Enter choice : ");
     scanf ("%d", &ch);
     switch (ch);
     {
       case 1: s = a+b;
      printf (" \n Sum is %d", s);
      break;
      case 2: s = a-b;
              printf ("\n difference is %d", s);
              break;
      case 3 : s = a* b;
              printf ("\n Product is %d", s);
              break;
      default : printf ("\n Exit");
      }
       getch();
  }
```

Date :
Page :

- Difference between if-else and switch statement.

If else statement.	Switch statement
1) In if-else selection appropriate option is done in serial fashion.	1) In switch case selection appropriate option is done in parallel way.
2) An expression is evaluated & the code block is selected based on the result of expression.	2) An expression is evaluated & code block is selected based on the value of expression.
3) It takes decision on the basis of true or false.	3) It takes decision on the basis of equality.
4) It doesnot require break statements since only one of the block of code is executed at a time.	4) It needs break statement to avoid execution of the block of code just below the currently executing block.

2) Repetitive / looping:

A loop is a repetitive control structure which execute a block of program statements repeatedly for specified number of time or till the given condition is satisfied. There are mainly three types of loop:-

a) while loop
b) do while
c) and for loop.

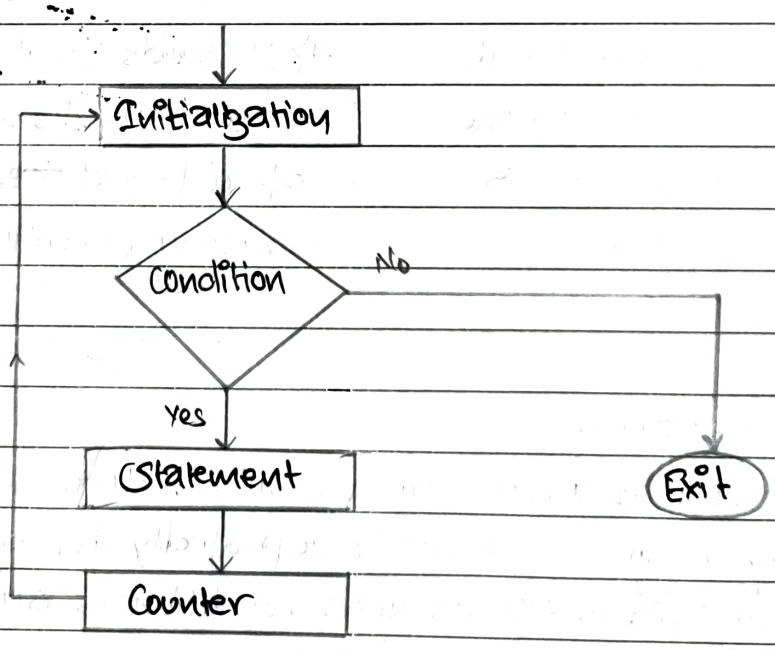
a) while loop:-

It executes a block statement until the given condition is true. It checks the condition at first, if it is true than it executes the statement otherwise it gets out from the ~~for~~ loop structure.

Syntax:-

```
initialization;
while (condition)
{
    statement;
    increment/decrement;
}
```

Flowchart:-



WAP to display user name 10 times

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int i=1 ; char n[10]
 clrscr();
 Printf("\n Enter a name:");
 Scanf("%c ",&n);
 while (i<=10)
 {
    printf("%c ", n);
     i++;
 }
  getch();
}
```

b) Do..... while loop:

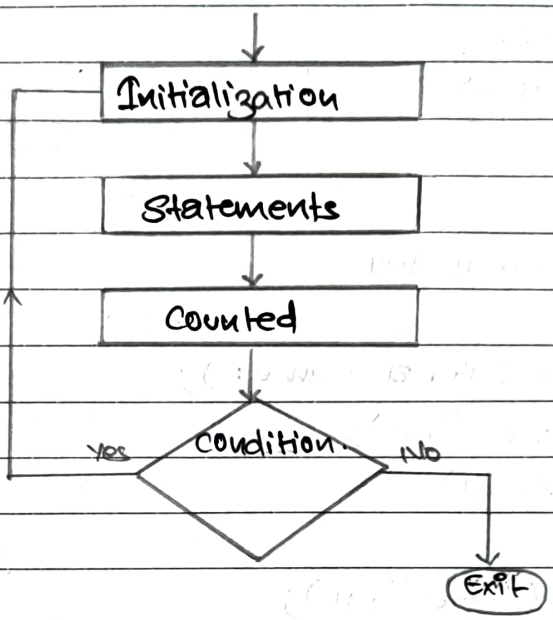
Do..... while loop executes the program statements until the given condition is true. Here, the program statements are executed at least once before the condition is checked. If the condition is found true it executes program statements again otherwise it gets out of the loop structure.

Syntax:

```
Initialization;
{
 statements;
  increment/decrement;
}
 while(condition);
```

Date :
Page :

Flowchart:



WAP to print Nepal 10 times.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i = 1;
  clrscr();
  do
   {
     printf("\n Nepal ");
      i++;
     }
     while (i< 10);
     getch();
}
```

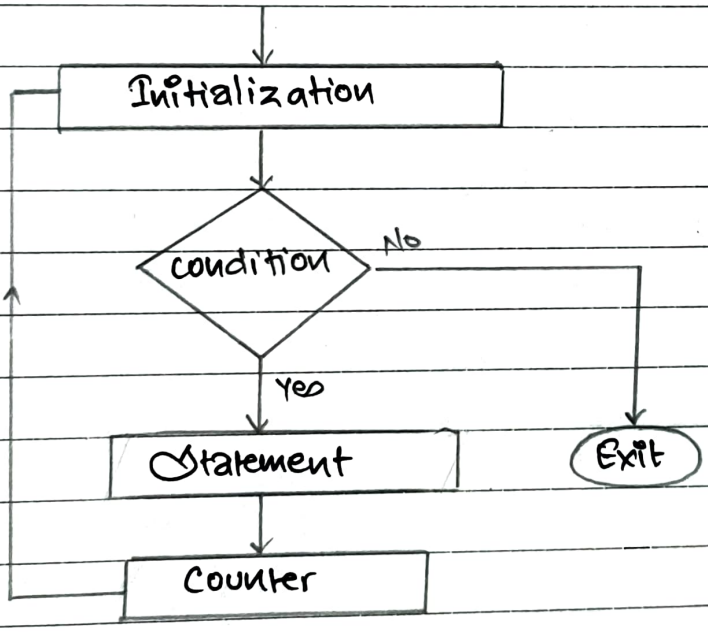
c) for loop:-

It is the most common type of loop that is used to execute the block of statements for a specified number of times. This is a looping statement that executes statements for required number of times.

Syntax:

```
for (initialization; condition; counter)
{
  statement;
}
```

Flowchart.



Date :
Page :

#WAP to print your name 10 times.

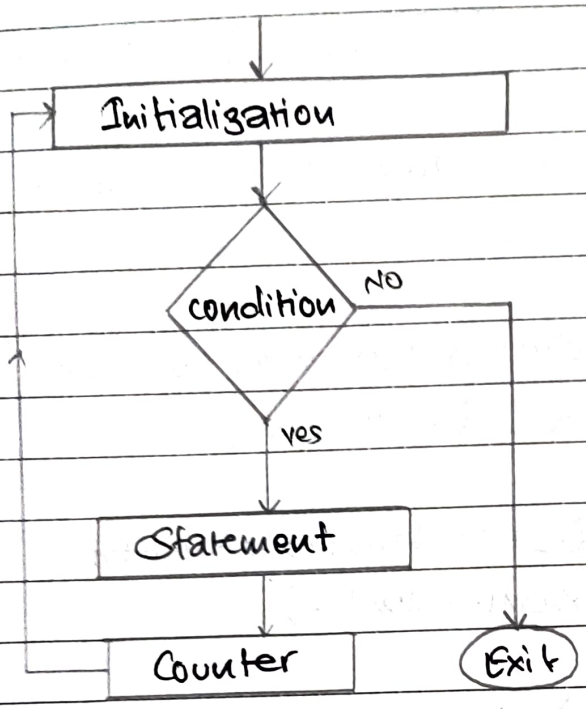
```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i;
  clrscr();
  for(i=1; i<=10; i++)
  {
    printf("\n Ram");
  }
  getch();
}
```

Difference between

while loop (Pre-test)	do while loop (Post test)
1) Condition is checked at beginning.	1) Condition is checked at end.
2) It is known as pre-test or entry control loop.	2) It is known as post test or exit control loop.
3) It is not terminated with semi-colon (;)	3) It is terminated with semi-colon (;)
4) Statement aren't executed if condition is false.	4) Statements are executed once even if condition is false.
5) It uses the keyword 'while'	5) It uses the keyword 'do' & 'while'

6) Syntax:
initialization;
while (condition)
{
statement;
increment/decrement;
}
	6) Syntax:
	initialization;
	do
	{
	statement;
	increment/decrement;
	}
	while (condition);

7) Flowchart:

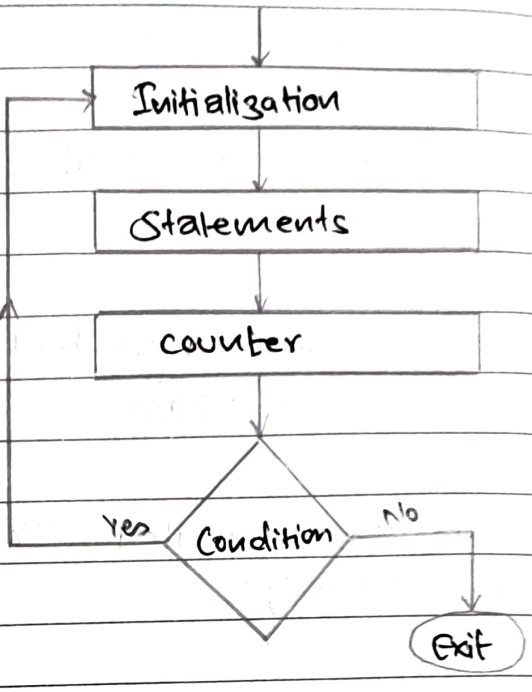


8) Example:

- WAP to display username 10 times.

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
     int i=1, charn[10]
     clrscr();
     printf("\n Entername: ");
     scanf("%c", &n);
    while(i<=10)
    {
      printf("%c", n);
       i++;
    }
     getch();
}
```

7) Flowchart;



8) Example:

- WAP to print Nepal 10 times.

```
#include<stdio.h>
#include <conio.h>
  void main()
  {
     int i=1;
     clrscr();
     do
      {
       printf("\n Nepal");
        i++;
      }
      while(i<=10);
      getch();
}
```

Infinite loop:-

A loop which never terminates is called infinite loop.

For eg:

```
for(i=0 ; i>=0; i++)
{
   printf("\n * ");
}
```

3) Jumping Statement:

→ Jumping statement are used to jump execution of program statements from one place to another. Following are the jumping statement at the c language :-

a) Break;
b) continue;
c) goto;
d) exit ;

a) break;

It is used to break the normal flow of the program execution in loop & switch case. When break is encountered in the program (loop body / switch statement) remaining part of the loop or switch is skipped.

Eg:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i;
  clrscr();
  for(i=0 ; i<=10 ; i++)
  →
```

```
{
    if (i==4)
        break;
        printf ("%d :", i);
}
getch();
}
```

b) Continue:-

When continue is encountered in the loop than that particular iteration is skipped & the loop will continue with the next iteration;

Eg:

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i;
  clrscr();
  for(i=0; i<=10; i++)
  {
    if (i==4)
    {
      continue;
    }
    printf ("%d ",i);
  }
getch();
}
```