

## Flowchart

initialization

condition

Statement

counter

Exit

#WAP to display user name 10 times.

#include <stdio.h>

#include <conio.h>

void main()

{

int i=1; char n[20];

clrscr();

while (i<=10)

{

printf("Enter name:");

scanf("%c", &n);

while (i<=10)

{

Print("%c", n);

i++;

g

getch();

g

## 5 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;

do

X

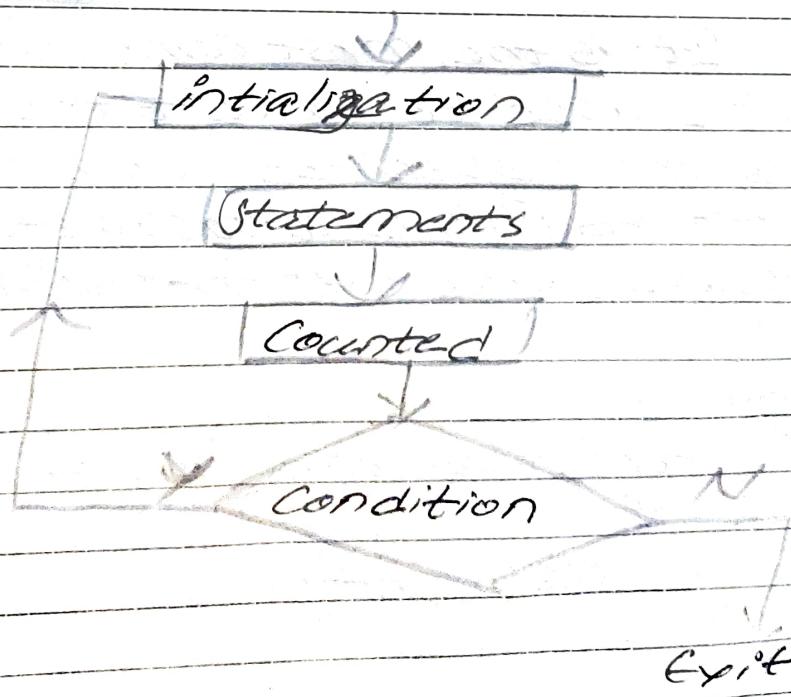
(statements);

increment / decrement;

g

while (condition);

### Flowchart



# WAP to print Nepal 10 times

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

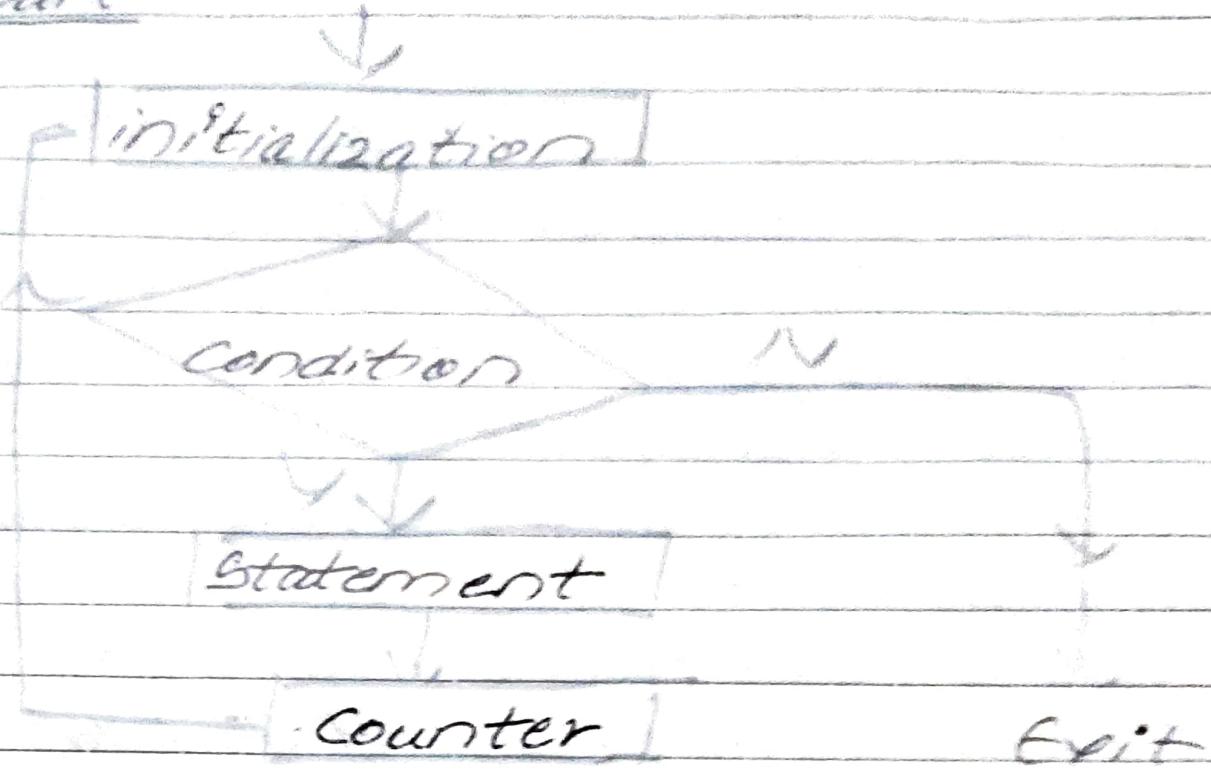
(c) for loop :-

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

Syntax

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

Flowchart



# 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");
    }
}
```

## Difference between

### 1. While loop

1. Condition is checked at beginning.
2. It is known as pre-test or entry control loop.
3. It is not terminated with semi-column ( ; )
4. Statement aren't executed if condition is false.
5. It uses the keyword `while`.

### 6. Syntax

```
initialization;
while (condition)
```

{

Statement;

in/creasement/decre;

}

### 7. Flowchart

### 8. Example

### do-while loop

1. Condition is checked at end.
2. It is known as post test or exit control loop.
3. It is terminated with semi-column ( ; )
4. Statements are executed once even if condition is false.
5. It uses two keyword `'do'` & `'while'`.

### 6. Syntax

```
initialization;
```

do

{

Statement;

in/creasement/decre;

}

while (condition);

### 7. Flowchart

### 8. Example

## # Infinite loop:-

A loop which never terminates is called infinite loop.

For E.g:-

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

## 3 Jumping statement:-

Jumping statement are used to jump execution of program statement 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.

E.g:- #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.

E.g:- #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();
```

### (c) Goto :-

It transfers the control of the program execution unconditionally to the location specified by goto.

E.g :- #include <stdio.h>

#include <conio.h>  
void main()

{

int i = 10;

label L:

printf("%d", i);

i++;

if (i <= 100)

    goto label L;

     getch();

}

### (d) Exit () :-

It is used for terminating the execution of program. This is a standard library function & uses the header file & <stdlib.h>.

Syntax :-

Exit (int - status)

status is the value returned to the operating system after the termination of program. The value 0 indicates termination is normal.

## Difference between

### Break

1. It terminates the block and the control goes out of the block.
2. It is a keyword that takes out the control from its containing block.
3. 

```
for(i=0; i<=10; i++)
```

```
{  
    if(i==3)  
        break;  
}
```

### Exit

1. It terminates the entire program & gives the control to the operating system.
2. It is a function that terminates the program.

```
...for (i=0; i<=10; i++)  
{  
    if (i==3)  
        exit(0);  
}
```

#WAP to display sum of all the digit  
of any number enter by user.

#include <stdio.h>

#include <conio.h>

Void main()

{

int n, sum=0, a;

clrscr();

printf("In enter the number :");

scanf("%d", &n);

While (n!=0)

{

a = ~~n~~ % 10.

sum = sum + a;

n = n / 10;

}

printf("In sum = %d", sum);

getch();

}