

Aim:- Program to learn iterative statements like while and do-while loops.

Software Required:- Turbo C/C++

Theory:-

while condition:- While loop statement in C programming language repeatedly executes a target statement as long as a given condition is true.

Syntax:-

~~while (condition)  
    {  
        statement(s);  
    }~~

do while condition:- A do-while loop is similar to a while loop, except that do-while loop is guaranteed to execute at least one time. The conditional expression appears at the end of the loop, so the statement(s) in the loop executes once before the condition is tested.

Syntax:-

```
do
    statement(s);
}
while (condition);
```

CODE :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i=1, n;
    clrscr();
    printf ("Enter number of terms: ");
    scanf ("%d", &n);
    while (i<=n)
    {
        printf ("%d\t", i);
        i++;
    }
    getch();
}
```

### Output of the CODE 1

Enter number of terms: 10

1 2 3 4 5 6 7 8 9 10

Q. #include <stdio.h>  
#include <conio.h>  
void main ()  
{  
int i=1, n, sum=0;  
clrscr();  
printf ("Enter Number of terms:");  
scanf ("%d", &n);  
while (i<=n)  
{  
sum = sum + i ; i++;  
}  
printf (" sum of first %d natural number = %d",  
sum);  
getch();  
}

Q. #include <stdio.h>  
#include <conio.h>  
void main ()  
{  
int i=1, n, f=1;  
clrscr();  
printf ("Enter a number :");  
scanf ("%d", &n);  
do  
{  
f = f \* i ; i++;  
} while (i<=n);  
printf (" Factorial of %d = %d ", n, f);  
getch();  
}

### Output of the CODE 2

Enter the number of terms : 10

Sum of first 10 natural numbers = 55

### Output of the code 3

Enter a number : 6

Factorial of 6 = 720

```
4. #include <stdio.h>
#include <conio.h>
void main ()
{
    int a=0, b=1, i=3, n, sum=0;
    clrscr();
    printf("Enter number of terms:");
    scanf("%d", &n);
    printf("FIBONACCI SERIES : ");
    printf("\n%d\t%d", a,b);
    do
    {
        sum = a+b;
        printf("\t%d", sum);
        a = b;
        b = sum;
        i++;
    } while (i<=n);
    getch();
}
```

## Output of the CODE 4

Enter number of terms : 10

0 1 1 2 3 5 8 13 21 34

## Viva Questions

1. How is while loop different from do-while loop?

Answer A "while" loop checks its condition before executing the loop body, meaning it might not run at all if the condition is initially false, while a "do-while" loop executes the loop body at least once because it checks the condition after the first execution, guaranteeing one iteration regardless of the condition.

2. What do you understand by ~~while(1) { /\* ... \*/ }~~ loop?

Answer A "~~while(1) { /\* ... \*/ }~~" loop is considered an "infinite loop" in programming, meaning it will continuously execute the code block within the curly braces as long as the program is running because the condition "1" is always true.

3. What do you understand by ~~while(0) { /\* ... \*/ }~~ loop?

Answer A "~~while(0) { /\* ... \*/ }~~" loop is essentially a "do nothing" loop in programming, as the condition "0" is always false, meaning the code inside the curly braces will never be executed.

4. What is the difference between pre-increment and post-increment operators?

The pre-increment operator in C is used to increment the value of any variable by one before assigning it to increment operator in C is used to increment the value of any variable by one after assigning it to another variable.

5. What changes should be made in the loop to stop it from running infinite times?

To stop a loop from running infinitely, you need to ensure the loop condition eventually becomes false by modifying the loop variable within the loop body, typically by incrementing it so that it reaches a point where the condition no longer holds true. You can also use a "break" statement to exit the loop permanently when a specific condition is met.

NKart  
8/10