IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Study of Loops In C Language

Shubham Sudhakar kshirsagar, Department of IT, GMVCS Tala, University of Mumbai
Tanuja Nitin Kasbale, Department of IT, GMVCS Tala, University of Mumbai
Ketaki Genaji Nadkar, Department of IT, GMVCS Tala, University of Mumbai
Madiha Murad Maner, Department of IT, GMVCS Tala, University of Mumbai

Prof.Raghvendra Singh
Assistant professor GMVCS & GMVIT
University of Mumbai

Abstract :- In this work, we conduct a systematic study of loops in C programs. We describe static analyses capable of efficiently identifying definite iteration in C code. One of the things computers can do more efficiently than humans is repetitive tasks. People get bored, and when that happens, their attention drifts and errors creep into the task at hand. Computers have the attention span of a gnat (i.e., none), so they are great at performing repetitive tasks. Unless a m c loses power or a component fails, they will loop forever, unless instructed to do otherwise. You may encounter situations, when a block of code needs to be executed several number of times.In general, statements are executed sequentially.The first statement in a function is executed first,followed by the second, and so on. Programming languages provide various control structures that allow for more complicated execution paths. A loop statement allows us to execute a statement or group of statements multiple times. A loop statement allows us to execute a statement or group of statements multiple times. Given below is the general form of a loop statement in most of the programming languages − C programming language provides the following types of loops to handle looping requirements.

Keywords :- Loops, loop pattern, structured programming, iteration, metrics, program study, ,statements, programming languages, function

I. INTRODUCTION

C programming language, there are circumstances where you want to do the same thing many times.You could type ten printf function, but it is easier to use a loop. The only thing you have to do is to setup a loop that execute the same printf function ten time**s.** A loop in c consist of two part, a body of a loop and a control statement. The control statement is a combination of some condition that direct the body of the loop to execute until the specified condition becomes false.The purpose of the c loop is to repeat the same code a number of time.C is ageneral-purpose programming language created by dennis Ritchie at Bell Laboratories in 1972.It is very popular language,despite being old. C is strongly associated with UNIX, as it was developed to write the UNIX operating system.C language it is one of the most popular programming language in the world.if you koow C, you will have no problem learning other popular programming language such as java, python, c+ +,C#,etc as the syntax is similar Historically, some programming languages provide restricted loop statements, such as the Fortran do statement or the Ada for statement. The for loop in C is a repetition control structure that aids in the creation of a loop thats runs a section of code repeatedly according to the loop's specifications.

What is looping:-

The looping can be defined as repeating the same process, multiple times until a specific condition statisfies. It isknown as iteration also. There are three loops type of loops used in c language. In this part of tutorials, we are going to learn all aspects of c loops.

What are loops in C:-

Loop is used to execute the block of code several times according to the condition given in the loop. It means it executes the same code multiple times so it saves code and also helps to traverse the elements of an array.

There are 3 types of loop –

•**While loop**

•**Do-While loop**

•**For loop**

1) While loop :- A "While" Loop is used to repeat a specific block of code an unknown number of times, until a condition is met. For example, if we want to ask a user for a number between 1 and 10, we don't know how many times the user may enter a larger number, so we keep asking "while the number is not between 1 and 10". If we (or the computer) knows exactly how many times to execute a section of code (such as shuffling a deck of cards) we use a for loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then and only then the body of a loop is executed. A while loop is the most straightforward looping structure. While loop syntax in C programming language is as follows:

Syntax of While Loop in C:

```
while (condition) {
        statements;
}
```

It is an entry-controlled loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then and only then the body of a loop is executed. After the body of a loop is executed then control again goes back at the beginning, and the condition is checked if it is true, the same process is executed until the condition becomes false. Once the condition becomes false, the control goes out of the loop.After exiting the loop, the control goes to the statements which are immediately after the loop. The body of a loop can contain more than one statement. If it contains only one statement, then the curly braces are not compulsory. It is a good practice though to use the curly braces even we have a single statement in the body.In while loop, if the condition is not true, then the body of a loop will not be executed, not even once. It is different in do while loop which we will see shortly.

2) Do- While loop:

A do…while loop in C is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop.
Syntax of do while loop in C programming language is as follows:

Syntax of Do-While Loop in C:

```
do {
  statements
} while (expression);
```

As we saw in a while loop, the body is executed if and only if the condition is true. In some cases, we have to execute a body of the loop at least once even if the condition is false. This type of operation can be achieved by using a do-while loop.
In the do-while loop, the body of a loop is always executed at least once. After the body is executed, then it checks the condition. If the condition is true, then it will again execute the body of a loop otherwise control is transferred out of the loop.
Similar to the while loop, once the control goes out of the loop the statements which are immediately after the loop is executed.
The critical difference between the while and do-while loop is that in while loop the while is written at the beginning. In do-while loop, the while condition is written at the end and terminates with a semi-colon (;)

3) For Loop:- The for loop in c language is used to iterate the statements or a part of the program several time.it is frequently used to traverse the data structures like the array and linked list. The syntrax of for loop in c language is given below;
for (Expression l ; Expression 2 ; Expression 3)

```
{ codes to be executed

}
```

Expression 1:-

- Represnt the initialization of the loop variable.
- More than one variable can be initialised.

Expression 2:-

- Expression 2 is a conditional expression.it cgecks for a specific condition to be satisfied If it is not, the loop is terminated.

- Expression 2 can have more than onecondition. However, the loop will iterate until the last last condition becomes false.other condition will be treated as statements.

Expression 3:-

- Expression 3 is increment or decrement to update the value of the loop variable.

When to Use a for Loop:-

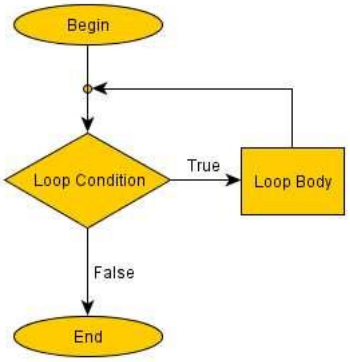
C provides you with several loop flavors, so how do you know which one to select? For the moment, let's just make a simple generalization and say: If you know how many passes are to be made through the loop before the loop begins execution, a for loop is usually a good choice. Another thing you'll like about for loops is that all three conditions for a well-behaved loop can be found within the parentheses of the for loop. expression1 usually is used to initialize the variable that controls the loop. expression2 usually involves a test that results in logic true or false , which determines whether another pass should be made through the loop. Finally, expression3 usually changes the state of the loop control variable. The syntax structure of the for loop makes room for all of the expressions to be in one place, almost forcing you to write a well-behaved loop. I'll have more to say about this topic after all loop structures have been discussed.

Condition And body:-

Every loop has a condition. This is the part that control if the loop should continue or stop. All loops in c continue their iteration if the condition is true. In c, the condition is any valid value or expression that could be evaluated to a value. A value of 0 or '\0' (null) os comsidered as "False" and everything else is "true" The body is one or more valid C statement. Is it consist of more than ome statement,the body must be enclosed in curly bracket{}.

How Loops in C work :-

The loop starts by entering the condition. If it is true the body will be executed. Then the execution repeats the check of the condition and then the body. It looks like this:
begin -> condition(true)->body-> condition(true)->body ... condition(false)->end.
One check of the condition, followed by one execution of the body is one iteration. Here is a flowchart to visualize the idea :-



II. METHODOLOGY

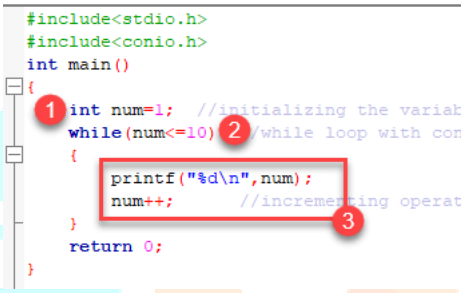
1. while loop in C example

```
#include<stdio.h>
#include<conio.h>
int main()
{
      int num=1;      //initializing the variable
      while(num<=10)  //while loop with condition
      {
            printf("%d\n",num);
            num++;          //incrementing operation
      }
      return 0;
}
```

Output:

1
2
3
4
5
6
7
8
9
10

The above program illustrates the use of while loop. In the above program, we have printed series of numbers from 1 to 10 using a while loop.



While loop in c programing

- We have initialized a variable called num with value 1. We are going to print from 1 to 10 hence the variable is initialized with value 1. If you want to print from 0, then assign the value 0 during initialization.
- In a while loop, we have provided a condition (num<=10), which means the loop will execute the body until the value of num becomes 10. After that, the loop will be terminated, and control will fall outside the loop.
- In the body of a loop, we have a print function to print our number and an increment operation to increment the value per execution of a loop. An initial value of num is 1, after the execution, it will become 2, and during the next execution, it will become 3. This process will continue until the value becomes 10 and then it will print the series on console and terminate the loop.
- \n is used for formatting purposes which means the value will be printed on a new line.

2. Do-while loop in C example

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num=1;      //initializing the variable
    do      //do-while loop
    {
        printf("%d\n",2*num);
        num++;          //incrementing operation
    }while(num<=10);
    return 0;
```

}

Output:

2

4

6

8

10

12

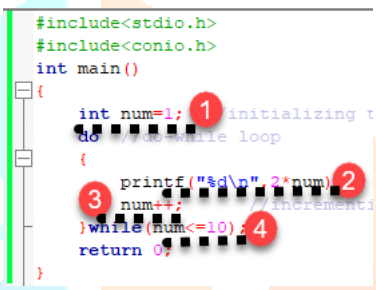
14

16

18

20

In the above example, we have printed multiplication table of 2 using a do-while loop. Let's see how the program was able to print the series.



Do-While Loop in C Programming

- First, we have initialized a variable 'num' with value 1. Then we have written a do-while loop.
- In a loop, we have a print function that will print the series by multiplying the value of num with 2.
- After each increment, the value of num will increase by 1, and it will be printed on the screen.
- Initially, the value of num is 1. In a body of a loop, the print function will be executed in this way: 2*num where num=1, then 2*1=2 hence the value two will be printed. This will go on until the value of num becomes 10. After that loop will be terminated and a statement which is immediately after the loop will be executed. In this case return 0

3. For loop in C example:

```
#include<stdio.h>
int main()
{
    int number;
    for(number=1;number<=10;number++)    //for loop to print 1-10 numbers
    {
        printf("%d\n",number);          //to print the number
    }
    return 0;
}
```

Output:

1

2

3

4

5

6

7

8

9

10

```c
#include<stdio.h>
int main()
{
    int number;
for(number=1;number<=10;number++)
    {
        printf("%d\n",number)
    }
    return 0;
}
```

The above program prints the number series from 1-10 using for loop.

For loop in C programing

- We have declared a variable of an int data type to store values.
- In for loop, in the initialization part, we have assigned value 1 to the variable number. In the condition part, we have specified our condition and then the increment part.
- In the body of a loop, we have a print function to print the numbers on a new line in the console. We have the value one stored in number, after the first iteration the value will be incremented, and it will become 2. Now the variable number has the value 2. The condition will be rechecked and since the condition is true loop will be executed, and it will print two on the screen. This loop will keep on executing until the value of the variable becomes 10. After that, the loop will be terminated, and a series of 1-10 will be printed on the screen

III. CONCLUSION

Hence we Had studied About loops in c programing.

REFERENCES:-

[1] A. S. Basarab, Klas LK-lup, Matematicheskie Issledovanija 120(1991), 3–7.
[2] R. Hubert Bruck, A Survey of Binary Systems, third printing, corrected, Ergebnisse der Mathematik und ihrer Grenzgebiete, Neue Folge 20, Springer-Verlag, 1971.