CL118 Programming Fundamentals

LAB 07

ITERATIVE STATEMENTS IN C

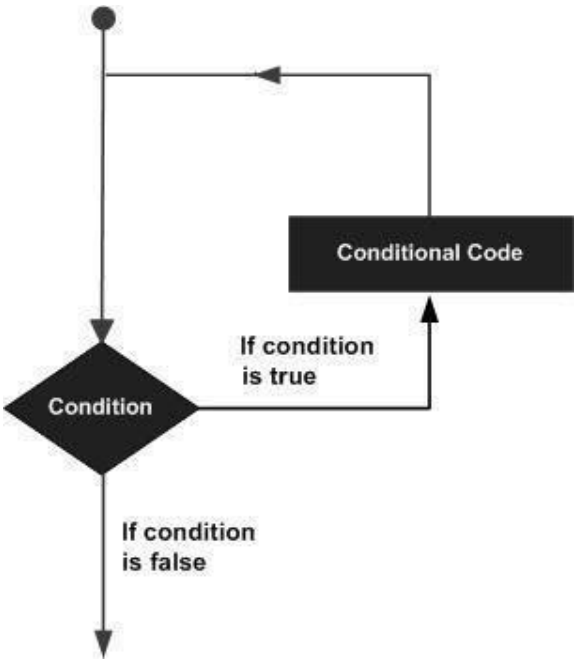
NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Learning Objectives

1. Iteration & iterative statements in C
2. While loop
3. Do-while loop
4. For loop
5. Continue Statement
6. Break Statement

1. Iteration and Types of Iterative C

By iteration or repetition, we mean doing the same task again and again. Usually, we want to repeat the task until some condition is met or for a predefined number of iterations. In computer science this is commonly referred as loop. Loops are used to repeat a statement, a block, a function or any combination of these. A typical looping workflow is shown in the figure.



Loops are very powerful and important tool in programming. Consider adding the salary of 1000 employees of a company. Or finding out how many people have been vaccinated from a list of 30 million people, that is just the population of Karachi what about whole Sindh or Pakistan? One way is to copy paste the code or repeating the lines of code. Well, the easier way is using a loop.

C programming language offers three kinds of iterative statements.

Types of Loops

1. While Loop
2. Do-while Loop
3. For Loop

2. While Loop

The while loop is usually used when we don't know the number of iterations in advance.

Application: As a game developer you don't have any knowledge how many times user will play the game so you want to loop over the choice if they want to play again or not.

a. C- Syntax

```
while ( loop repetition condition )
{

        Body of the loop.

 }
```

b. Interpretation:

- The loop repetition condition (a condition to control the loop process) is tested; if it is true, the statement (loop body) is executed, and the loop repetition condition is retested.
- The statement is repeated as long as (while) the loop repetition condition is true. When this condition is tested and found to be false, the while loop is exited and the next program statement after the while statement is executed. If the condition is true forever the loop will run forever, we call such loop an infinite loop.
- It may not be executed at all if condition is false right from start.

c. Example:

To count number of digits in a given integer.

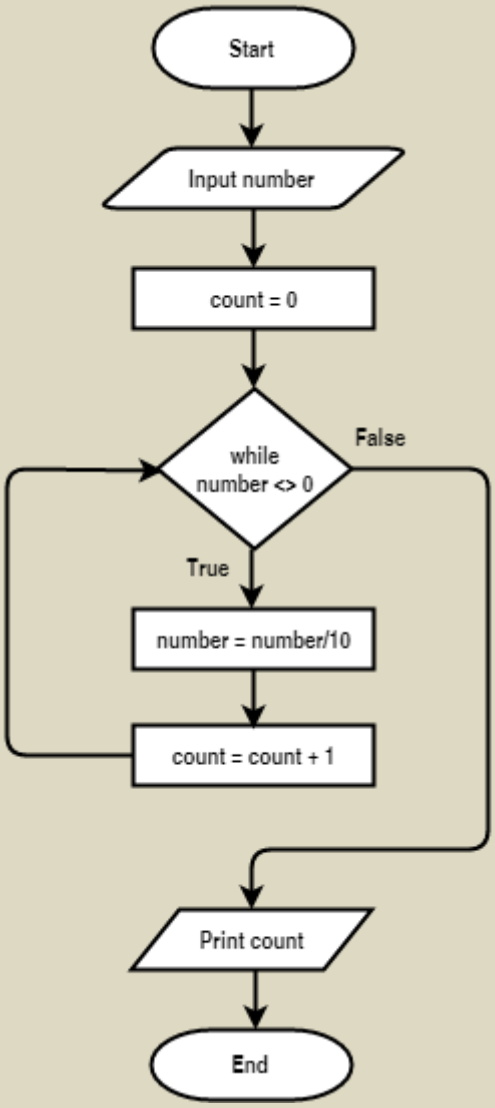
Input: 13542

Output: 5

ALGORITHM

1. Start
2. Input number
3. count = 0
4. While number != 0 do
5. number=number/10
6. count = count + 1
7. While End
8. Print count
9. End

FLOWCHART



C-IMPLEMENTATION

```
#include <stdio.h>

    int main()
    {
        int number=0, count =0;
        scanf("%d",&number);
        while (number != 0)
        {       number = number/10;
                count = count + 1;
        }
        printf("The number of digits are: %d", count);
        return 0;
}
```

```
while ( loop repetition condition )
{
        Statement_Block;
}
```

3. Do-while Loop

The do-while loop checks the loop repetition condition after running the body of the loop. This structure makes it most favorable in conditions where we are interested in running the body at least once.

Application: As a web-developer you are writing a registration application. You would like to make sure if the username is already taken or not. Therefore, you will keep asking for a unique username until provided, in such situations you would want the user to provide the username first and then perform the check.

a. C- Syntax

```
do
{

        Body of the Loop;

} while ( loop repetition condition );
```

b. Interpretation

- First, the body of the loop is executed.
- Then, the loop repetition condition is tested if it is true, the body is again and the condition is retested until it remains true the loop continues. When this condition is tested and found to be false, the loop is exited and the next statement after the do-while is executed.

c. Example

Repeating the same example with do-while. Count number of digits in a given integer.

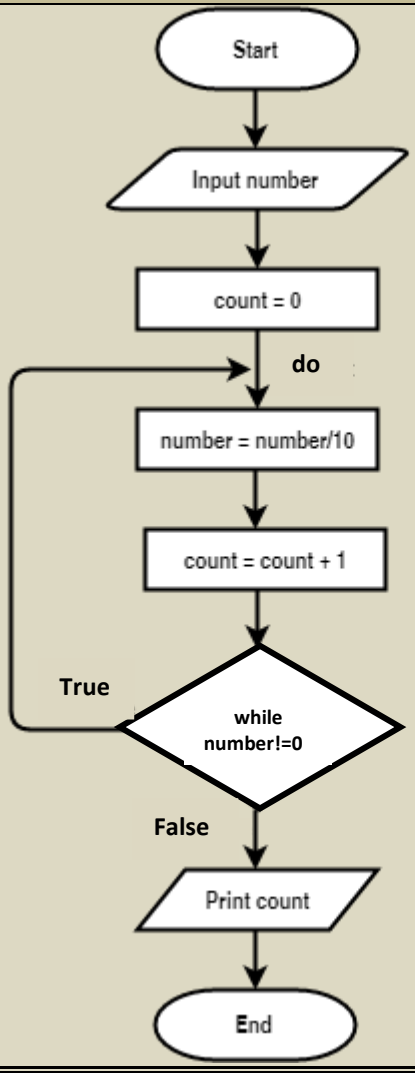
Input: 144452

Output: 6

ALGORITHM

1. Start
2. Input number
3. count = 0
4. do
5. number=number/10
6. count = count + 1
7. while number != 0
8. Print count
9. End

FLOWCHART



C-IMPLEMENTATION

```
#include <stdio.h>

int main()
{
        int number=0, count =0;
        scanf("%d", &number);
        do
        {
                number = number/10;
                count = count + 1;
        } while (number != 0);

        printf("The number of digits are: %d", count);
        return 0;
}
```

```
do
{
        Statement_Block;

} while ( loop repetition condition );
```

4. For Loop

The **For** loop is mostly used when we want to iterate for a specific number of times.

Application: You are developing a first-person shooter game in the beginning of the mission user is given 3 lives only, you would like the character to revive if killed for 3 times only.

a. C- Syntax

```
for ( initialization expression ; loop repetition condition ; update expression )
{
        Body of the Loop;
}
```

b. Interpretation

- The *initialization* is an assignment statement that is used to initialize the loop control variable with a value. This is the first statement to be executed in the loop and only run once.
- The *condition* is a relational expression that determines when the loop exits. This runs after initialization, and verified before every iteration.
- The update expression either *increment* or *decrement* the loop control variable on each iteration.

c. Example

Print the numbers from 0 to desired value as shown below.

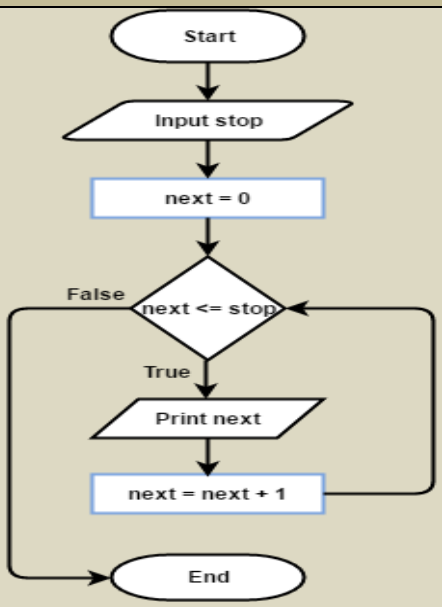
Input: 10

Output: 0 1 2 3 4 5 6 7 8 9 10

ALGORITHM

1. Start
2. Input stop
3. FOR next = 0 to stop
4. Print next
5. next = next + 1
6. END FOR
7. End

FLOWCHART



C-IMPLEMENTATION

```
#include <stdio.h>

int main ()
{
int stop, next;
printf("Enter ending value:");
scanf("%d",&stop);

  for(next = 0 ; next <= stop ; next=next+1)


{
    printf("%d\t", next);
  }

  return 0;
}
```

```
for (initialization expression ;
loop repetition condition ;
update expression )
{
        Statement_Block;
}
```

OUTPUT

Comparison of Loops

	for	while	do-while
Initialization of condition variable	In the parenthesis of the loop.	Before the loop.	Before the loop or in the body of the loop.
Test condition	Before the body of the loop.	Before the body of the loop.	After the body of the loop.
Updating the condition variable	After the first execution.	After the first execution.	After the first execution.
When to use	for is generally used when there is a specific number of iterations	while is generally used when the number of iterations is not known in advance.	do-while is a loop with a post-condition. It is needed in cases, when the loop body is to be executed at least once.

5. Continue Statement

The continue statement is very powerful in situations where we want to execute some portion of the loop body and skip a statement or block of statements. The control skips the loop body as it reaches the continue statement and starts the next iteration. Usually there is a condition for which we want to skip certain statements.

Application: As an AI developer you would work with a lot of datasets (e.g., thousands of images). Before training your AI model on these datasets you need to filter the corrupted or invalid images from the valid once. In such situations you can utilize continue on dependent situations instead of using multiple conditional statements.

a. C Syntax

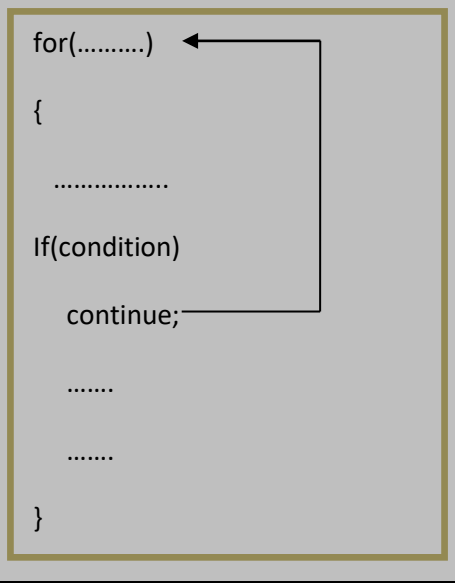
```
for (initialization; condition; increment/decrement) {

    block of statements;     // this block will execute always with each iteration of loop
    continue;
    block of statements;     // this block will be skipped.

}
```

b. Example:

```
#include<stdio.h>
int main()
{
   int i;
   for (int i=0; i<=10 ;i++)
   {
     if((i==3)||(i==7))
     {
          continue;
     }
      printf("The sum is %d", sum);
     }
   }
```



6. Break Statement

Break statement is used to exit the body of the loop without meeting the loop repetition condition. The statements after the break never gets executed and usually break is used to stop the loop before the loop termination condition is met. The controls execute the next statement after the loop body once it reaches the break statement in the loop body. Usually there is a condition upon which we want to exit the loop.

Application: You have written a fund-raising application for a cancer patient, after the required funds are collected, you would like to break out of the loop without knowing how many iterations have passed.

a. C Syntax

```
for (initialization; condition; increment/decrement) {

    block of statements;     // this block will execute always with each iteration of loop
    break;
    block of statements;     // this block never gets executed.

}
```

b. Example

```
#include<stdio.h>
int main()
{
   int a, sum=0;
   for(;  ;)
   {
     scanf("%d",&a);
     if(a==-999)
     {
     break;
     sum=sum+a;
     }
     }
printf("The sum is %d", sum);
```

```
for(…….)

{

  ……..

If(condition)

    break;

    …….

    …….

}
printf("…");
```

LAB 05 EXERCISES

INSTRUCTIONS:

NOTE: Violation of any of the following instructions may lead to the cancellation of your submission.

1) Create a folder and name it by your student id (k21-1234).
2) Paste the .c file for each question with the names such as Q1.c, Q2.c and so on into that folder.
3) Submit the zipped folder on slate.

ARRAYS ARE NOT ALLOWED TO BE USED FOR SOLVING THE FOLLOWING EXERCISES. DO NOT USE BREAK OR CONTINUE STATEMENTS INSIDE LOOP.

QUESTION# 01

Write a program which will find the factorial of a given number. Exit the program if the input number is negative.

Example of Factorial: Input number = 5 Factorial is=5*4*3*2*1

Note: Justify your choice of loop answering two important points: why your choice is optimal? why other looping structure would not be suitable?

QUESTION# 02

Ask the user to input an integer (N > 10) then input the N numbers from user only allowing positive integers. Calculate the average value of the inputs, count the odd, even and the numbers which are factor of 3

Note: Perform the task with do-while loop.

QUESTION# 03

Write a program which will input a 5-digit number. If the sum of digits is even, find whether the input number is a prime or not. If the sum of digits is odd find, whether the number is palindrome or not?

Example of prime number: A number which is only divisible by itself and 1 i.e., 7, 11, and13.
Example of a Palindrome: A number whose reverse order is the same as the original number i.e., 11211, 44344.

QUESTION# 04

Write a program which will generate the Fibonacci series up to 1000. Also find the sum of the generated Fibonacci numbers divisible by 3, 5 and 7 only.

Example of Fibonacci series is: 1 1 2 3 5 8 13 25.....

Note: Do this task by using ***for loop*** and ***while loop***. Also identify which one is more efficient?

QUESTION# 05

Write a program which inputs the 25 student's marks (out of 100) and counts the number of failed and passed students. How many students are excellent and how many are above average.

Note: Passing marks = 50, A+ = 90 – 100 (Outstanding), A = 86 - 89 (Excellent), B+ = 75 – 85 (Good), B- = 70 – 74 (Average), C+ = 66 – 69 (Below Average), C = 62 – 65 (Adequate), C- = 58 – 61 (Pass), D+ 54 – 57 (Pass), D = 50 – 53 (Pass).

QUESTION# 06

You are requested to write a program which will be used to take the entry test for university students. The program must have following features.

- For correct answer, students get 4 marks.
- For wrong answer, student lose 1 mark.
- If the student answers first four questions wrong exit the program with a message "Sorry, you did not qualify for the admission."
- If students score 20 marks, program should display "Congratulations, you have qualified for the admission "and exit.
- There will be only 4 options for each question

QUESTION# 07

You are requested to write an application which ask the user to input 3 alphabets of their choice. After that continuously ask the user for new alphabet, if all three alphabets of choice are entered either in a forward sequence or backward, the application ends with telling them how many times the user has entered each alphabet of the choice.

Note: you are asking one alphabet at a time, and only terminate the application if the alphabets were provided in a sequence.

QUESTION# 08

Develop a user-registration system have the following options.

a. Ask the user for a user-name (5 alphabets).
b. Password should be 6 characters long with at least 1 numeric, 1 capital and 1 small letter.
c. Display a "Account Created Successfully".
d. Login the user with correct username and password.
e. Display "Welcome username, you are now logged in".

Note: Develop your application using break and continue statement.

QUESTION# 09

At a popular car showroom, a salesperson is rewarded 2% of the sales s/he do. If a salesperson makes the sales worth of 10 million their reward is increased to 3.5%. Calculate the net salary of all the salesperson at the showroom by asking their base salary and monthly sales. No salesperson has base salary less than RS. 60,000 exit the program if base salary is less than RS. 60,000.

QUESTION# 10

Your city bank has revised their credit card policy and have requested you to write a C program which will identify if a user has exceeded their credit limit or not. The program should ask the following information from the user

a) Account number

b) Balance at the beginning of the month
c) List of all items purchased this month along with their charges, calculate the total charges
d) Total of all credits applied to this customer's account this month
e) Allowed credit limit

The program should input each fact, calculate the new balance = (beginning balance + total charges – credits), and determine whether the new balance exceeds the customer's credit limit. For the customer whose credit limit is exceeded, the program should display the customer's account number, credit limit, new balance and the message “Credit limit exceeded by 123.74.”
All the users have purchased at least one product. You have the liberty to ask the user how many products they want to enter.

Following are the sample input/output dialogs:

```
Enter account number (-1 to end): 100
Enter beginning balance: 4568.78
How many products you would like to Enter: 5
Enter charges for item 1: 150.00
Enter charges for item 2: 200.00
Enter the charges for item 3: 100.50
Enter the charges for item 4: 150.50

Enter the charges for item 5: 500.75
Your total: 1101.75
Enter total credits: 500.00
Enter credit limit: 5000.00

Account: 100
Credit limit: 5000.00
Balance: 4179.53
Credit Limit Not Exceeded.
```