



Digital Design Verification

Lab # 03

LOOPS, Problem Solving

Release: 1.1

Date: 30-July-2024

NUST Chip Design Centre (NCDC), Islamabad, Pakistan

Copyrights ©, NUST Chip Design Centre (NCDC). All Rights Reserved. This document is prepared by NCDC and is for intended recipients only. It is not allowed to copy, modify, distribute or share, in part or full, without the consent of NCDC officials.

Revision History

Revision Number	Revision Date	Nature of Revision	Approved By
1.0	2/05/2024	Complete manual	Dr. Abid
1.1	30/07/2024	Revision in manual	Engr Hira



Contents

OBJECTIVE:.....	4
TOOLS:.....	4
EXAMPLE	4
TASK # 01:.....	5
TASK # 02:.....	5
TASK # 03:.....	6
TASK # 05:.....	7
TASK # 06:.....	8
SUBMISSION:	9



[Azad's Law: The one facilitating cheating will be punished with 0 marks]

OBJECTIVE:

The objective of this lab is to enable students to answer following questions:

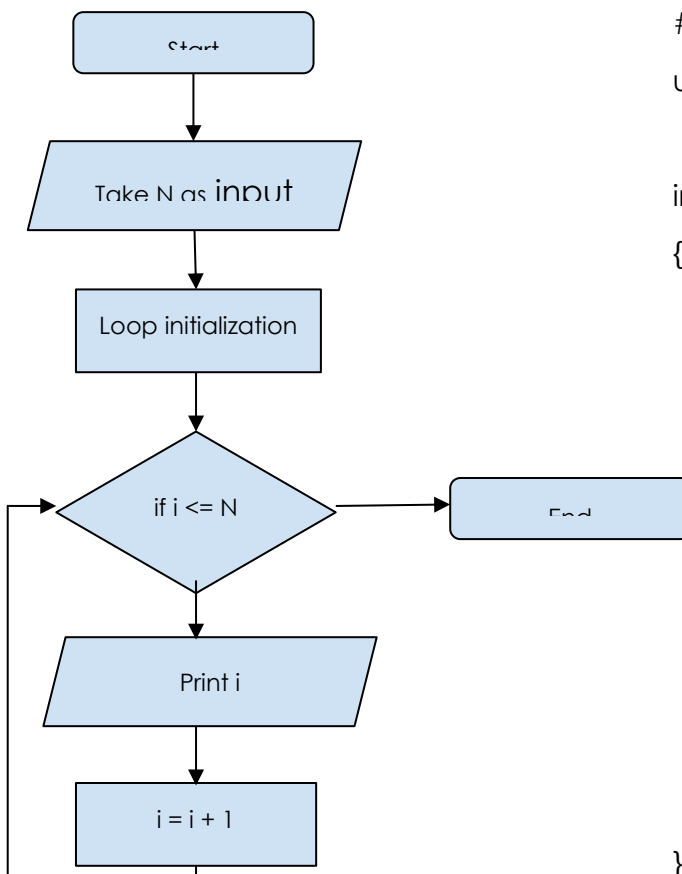
- Why are iterative algorithms used for solving a problem?
- How does C/C++ support these algorithms using loops?
- Dissection of loop into distinct components; initialization, termination and increment
- How to use flow charts to capture iterative algorithms?
- What is the difference between for and while loop and when to use each of them?
- How to apply these concepts to real-world applications.
- How to use nested loops

TOOLS:

- GCC
- GDB

EXAMPLE

If you want to do something repetitively, you can do it effectively using loops. Consider an example where one has to write a program to print 1,2,3,,N integers where N is the input taken from the user.



```
#include <iostream>
using namespace std;

int main()
{
    cout << "Enter the integer
N"<<endl;
    cin >> N;

    for (i = 1; i<= N; i++)
    {
        cout << i << endl;
    }

    return 0;
}
```



TASK # 01:

Cricket Statistics - There is cricket fever out there everywhere in Pakistan. Consider Babar Azam statistics for the last few world cups. Take the number of matches he has played in world cups. Now take the runs scored by Babar Azam in each match along with the balls he faced. At the end your program should print the following for Babar Azam:

- strike rate,
- batting average
- minimum score
- maximum score
- number of centuries

Flow Chart : 20 minutes **Coding** : 20 minutes

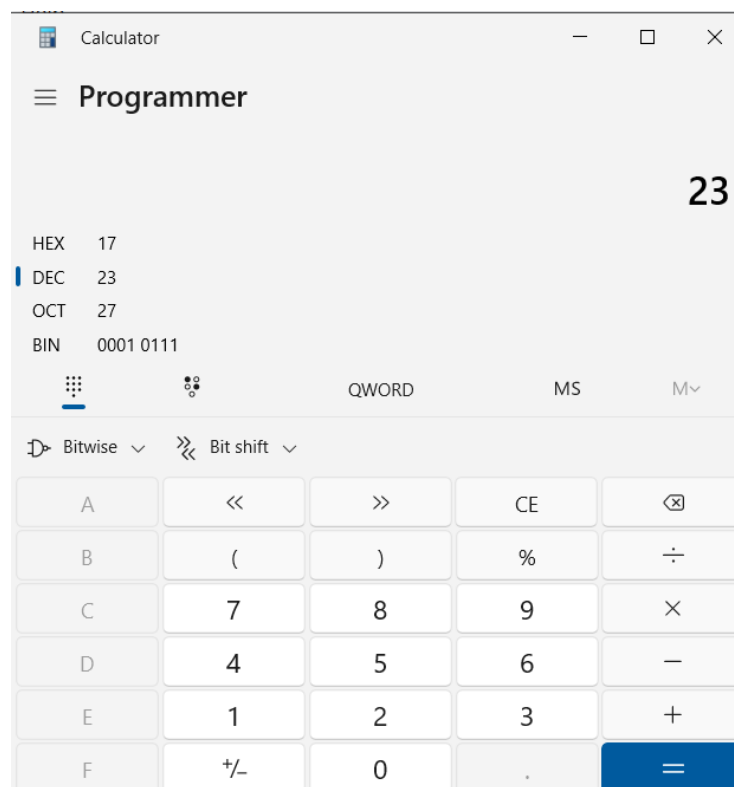
TASK # 02:

Number Converter

Take a number N in decimal format from the user and convert it into

- Hexadecimal format
- Binary Format
- Octal Format

Example: N = 23

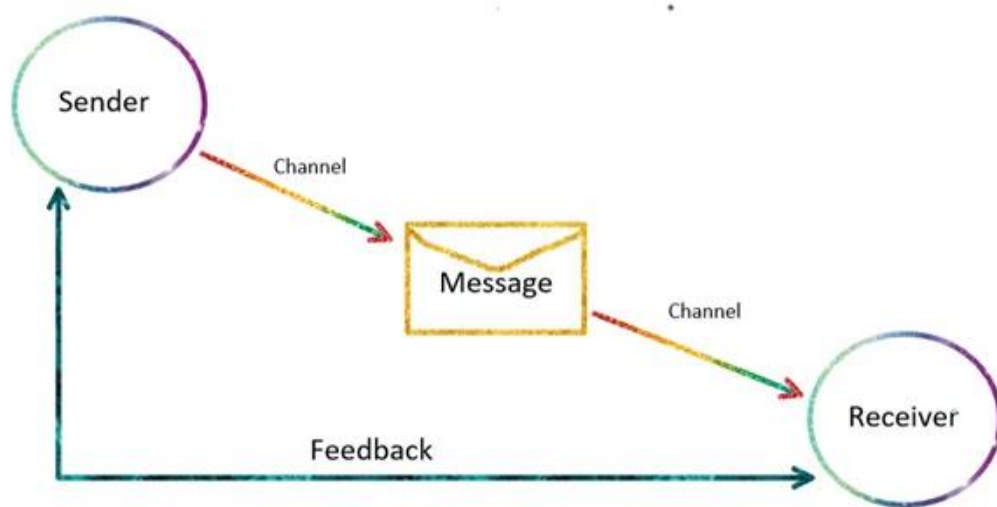




Flow Chart: 15 Minutes **Code:** 15 Minutes

TASK # 03:

Imagine a communication system where a sender sends some message in the form of a data packet to a remote receiver. In order to ensure that the correct packet reaches a remote destination, the sender generates a checksum of the data and appends it with the packet. Upon reception, the receiver also generates the checksum using the same algorithm and matches with the checksum sent by the sender. If both match, it means data is correct.



- Take the number of bytes as input from the user
- Take the byte one by one from the user
- Generate the checksum by computing the **XOR** of all the bytes. You can take XOR of a and b using **a ^ b**.

Flow Chart: 25 Minutes **Code:** 25 Minutes

TASK # 04:

Number Guesser

Ask the computer to generate a random number N using the following code snippet

```
#include <cstdlib>
```



```
#include <iostream>

using namespace std;

int main()
{
    int Limit = 100;

    int N = rand() % Limit; // Will generate a number between 0 and Limit - 1

    return 0;
}
```

Modify the code to ask the user to keep guessing the number. If the guess is less than the number N, inform the user “The number is less than N” and then ask him to enter the number again. If the guess is greater than the number N, then inform the user “The number is greater than N”. Keep doing it until the number becomes equal to the number N.

Flow Chart: 15 Minutes **Code:** 15 Minutes

TASK # 05:

Imagine you are playing a grid (4x4) game where the player can move in four directions Up, Down, Right and Left. Take the initial position of the player from the user e.g. (1,2). Now take the direction in which the player will move and keep doing it until it collides with the boundary. Figure out whether it collides with the boundary or not? If it collides, it should simply exit by printing the coordinate of the boundary in the form “The player has reached the boundary at (x,y)” where x,y will be the coordinates of the boundary.

(1,1)			(4,1)
(1,4)			(4,4)

Flow Chart: 15 minutes

Coding: 25 minutes



TASK # 06:

Stopwatch Design: Design a stop watch which counts the number of minutes (N). Ask the user to enter the number of minutes.

Note: The watch should display the counter like this. If N = 2, it should count up to 2 minutes.

M : S (M = Minutes, S= Seconds)

00 : 00

00 : 01

00 : 02

00 : 03

|

|

|

00 : 59

01 : 00

01 : 01

|

|

|

01 : 59

TASK # 07:

Treasure Hunt: Imagine there is a ground which is represented by a 2D grid as shown below. There is a treasure hidden in one of the locations. You have to figure out where exactly the treasure is.

- Your program will not take any input from the user.
- It will generate the location of the hidden treasure (xt,yt) using a random number. You have used random numbers in the previous lab. Choose appropriate values to generate random location within the grid.



- Now search the grid using loops to figure out what is the location of the hidden treasure. Once you found it, print “Hurrah!, I have found the hidden treasure”.

(1,1)			(4,1)
(1,4)			(4,4)

Flow Chart: 25 Minutes **Code:** 25 Minutes

SUBMISSION:

Please submit .c files of all the tasks along with the screenshots of outputs on LMS in a proper report. Add snaps of all the flow charts in your report.