# Green University of Bangladesh

**Department of Computer Science and Engineering (CSE)**
Faculty of Sciences and Engineering
Semester: Spring, Year: 2022, B.Sc. in CSE (DAY)

**LAB REPORT NO # 05**

**Course Title: Structured Programming Lab**
**Course Code: CSE 104** **Section: CSE 213 - DB (PC)**

**Lab Experiment Name(s):**

- Lab Report of Problem-Solving Using Loops in C

**Student Details**

| Name | ID |
|---|---|
| Md. Shahidul Islam Prodhan | 213902017 |

**Lab Date:** 05 March 2022

**Submission Date:** 12 March 2022

**Course Teacher's Name:** Md. Solaiman Mia, Assistant Professor.

**[For Teacher's use only: Don't write anything inside this box]**

**Lab Report Status**

| **Marks:** | **Signature:** |
|---|---|
| **Comments:** | **Date:** |

## 1. TITLE OF THE LAB EXPERIMENT

Lab Report of Problem-Solving Using Loops in C

## 2. OBJECTIVES

Doing this experiment we will learn about loops in the C language and we will be able to solve complex problem using loops.

## 3. PROCEDURE

***Problem 1: Check if number is prime or not from existing algorithm.***

First we declare the necessary variables in int function and take input from the user a valid integer by using printf and scanf function. We use mod to check if the remainder is 2 or 0. If the count is 2 it is a prime number otherwise it is not a prime number.

***Problem 2: All Fibonacci numbers bellow n***

First we declare necessary variables and take valid input from the user. We take user input for necessary nth terms. The logic of Fibonacci is – the 3$^{rd}$ number will be the addition of 1$^{st}$ and 2$^{nd}$ number the loop will going on until the number is equal or less than nth terms.

***Problem 3: Pascal's triangle until given row.***

At first, we declare necessary variables and take valid user input. We use for loop in. Then we use nested loop to solve this problem.

## 4. IMPLEMENTATION & TEST RESULT

***Problem 1: Check if number is prime or not from existing algorithm.***

```
# include <stdio.h>
# include <stdlibh>

int main ( )
{
   int n, i, count = 0;
   printf ("Enter a number : \n");
   scanf ("%d", &n);

  for (int i = 1; i<=n; i++){
   if (n%. i== 0) {
        count ++;
      }
  }
  if (count ==2) {
    printf ("%d is a prime number.", n);
    }
  else { printf ("%d is not a prime
                  number. ", n);
      }
  return 0;
}
```

```
"C:\Users\shahi\OneDrive\Documents\Test Code\prime or not.exe"
Enter a number:
47
47 is a prime number.
Process returned 0 (0x0)   execution time : 1.766 s
Press any key to continue.
```

***Problem 2: All Fibonacci numbers bellow n***

```
# include <stdio.h>
int main()
{
  int num1 = 0, num2 = 1, num3, count, n_terms;
printf ("Enter the number of terms to
      be printed ! \n");
scanf ("%d", &n_terms);
printf ("\n%d \n%d \n", num1, num2);
for (count 3; count<=n_terms; count ++)
{
   num3 = num1 + num2;
   printf ("%d\n", num3);
   num1 = num2
   num2 = num3;
   }
return 0;
}
```

```
"C:\Users\shahi\OneDrive\Documents\Test Code\fibonacci series.exe"
Enter the number of terms to be printed:
10

0
1
1
2
3
5
8
13
21
34

Process returned 0 (0x0)   execution time : 1.366 s
Press any key to continue.
```

***Problem 3: Pascal's triangle until given row.***

```
# include <stdio.h>
 int main()
{
  int rows, col=1, space, i, j;
  printf ("Enter the number of rows: ");
  scanf ("%d", &rows);
  for (i=0; i<rows; i++) {
     for (space=1; space<=rows-i; space++)
      printf("  ");
     for (j=0; j<=i; j++) {
       if (j==0 || i==0)
         col=1;
       else
           col=col*(i-j+1)/j;
      printf ("%4d", col);
      }
     printf ("\n");
     }
  return 0;
}
```

```
"C:\Users\shahi\OneDrive\Documents\Test Code\pascals triangle.exe"
Enter the number of rows: 5
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1

Process returned 0 (0x0)   execution time : 2.433 s
Press any key to continue.
```

## 6. ANALYSIS AND DISCUSSION

1) We have solved those problems using CodeBlocks IDE and there were no errors occurred. And we can successfully print the output of those problems.
2) We have faced a little bit difficulty while performing pascals problem showing the wrong results but then we have passed errors and corrected our program and it gives correct result.
3) Solving these 3 problems, we have initially learned some of the advanced level of complex problems in C language using different loops.

## 7. SUMMARY

From the given experiments, we have learned the use of different loops in various complex problems and their usage varies from problem to problem and how to use some advance complex problems using loops.