



National University of Computer and Emerging Sciences, Lahore Campus

Programming Fundamentals

Assignment 3

Total Marks: to be decided by TA

Section: BSE-1A, BSE-1B

Due Date: 13th October 2022

Instructions:

Please read the following instructions carefully before submitting the assignment.

1. Follow TA's instructions for submission
2. It should be clear that your assignment will not get any credit if:
 - The assignment is submitted after due date.
 - Assignment is copied (partial or full) from any source (websites, forums, students, etc)

Objective:

The objective of this assignment is to provide an on-hand experience of:

1. Learn to use relational and logical operators in C++ program.
2. Learn to use of loops, nested loops

Guidelines:

1. Code should properly be indented and well commented.
2. Follow C/C++ naming conventions while using variables etc.

Assignment Description:

TASK 1:

It is known from Babylonian times that the square root of a number can be approximated by a method now known as 'divide and average' method. Since the Babylonians used it, it is also called the Babylonian algorithm.

METHOD: Suppose we want to compute the square root of a number N .

Let $A > 0$ be a guess for square root (N), then a better approximation is given by: $B = (A + N/A)/2$.

We can then improve the approximation B using $C = (B + N/B)/2$.

EXAMPLE: For example, let's compute: square root of $N=2$.

Let our initial guess be 1.

The next better approximation is $(1 + 2/1)/2 = 1.5$

The next better approximation is $(1.5 + 2/1.5)/2 = 1.416667$

The next better approximation is $(1.416667 + 2/1.416667)/2 = 1.414216$

And so on. The more you repeat this, the closer you will get to the actual answer. Just keep in mind that the square root of 2 is an irrational number, which means that you can keep on improving your approximation forever.

1. Write a C++ Program that implements above-mentioned method of calculating square root and will return the calculated square root. You have to run 10 iterations for approximation.
2. Keep repeating the code until user enters a -1 for terminating the program

TASK 2:

Write a C++ program for the following:

The factorial of a number is the product of all the integers from 1 to that number. For example, the factorial of 6, $6! = 1*2*3*4*5*6 = 720$. Factorial is not defined for negative numbers, and the factorial of zero is one, $0! = 1$

2. Write code which will take two inputs and find nPr (number of permutations).

$$nPr(n, r) = n! / (n-r)!$$

3. Write code which will take two inputs and find nCr (number of combinations)

$$nCr(n, r) = n! / (n-r)! * r!$$

4. Write a function which will take two inputs and find nCr (number of combinations) where

$$nCr(n,r) = nPr(n,r) / r!$$

5. Design a menu function that will take a number from the user as input and then perform specific operation based on input value.

Sample Output:

- a. On '1' print factorial of number.
- b. On '2' print nPr.
- c. On '3' print nCr designed in part 3.
- d. On '4' print nCr designed in part 4.
- e. On '5' Exit Program.

Draw following Patterns for positive values of integers provided by user as input.

TASK 4:

$$F_0 = 0, \quad F_1 = 1$$

and $F_n = F_{n-1} + F_{n-2} \quad n > 1$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

1. Display n^{th} term in the Fibonacci sequence
2. Display first n Fibonacci numbers
3. Display Fibonacci numbers between m and n

A sample run for **case 1** is given below:

```
What do you want to do?
  1. Display nth term in the Fibonacci sequence
  2. Display first n Fibonacci numbers
  3. Display Fibonacci numbers between m and n

Enter your choice: 1

Enter n (term number): 11

11th term in the Fibonacci sequence is 55
Press y if you want to continue again?
Y
What do you want to do?
  1. Display nth term in the Fibonacci sequence
  2. Display first n Fibonacci numbers
  3. Display Fibonacci numbers between m and n
.....
```

A sample run for **case 2** is given below:

```
What do you want to do?
  1. Display nth term in the Fibonacci sequence
  2. Display first n Fibonacci numbers
  3. Display Fibonacci numbers between m and n

Enter your choice: 2

Enter n (number of terms): 9

First 9 Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, 21

Press y if you want to continue again?
Y
What do you want to do?
  1. Display nth term in the Fibonacci sequence
  2. Display first n Fibonacci numbers
  3. Display Fibonacci numbers between m and n
.....
```

A sample run for **case 3** is given below:

What do you want to do?

1. Display nth term in the Fibonacci sequence
2. Display first n Fibonacci numbers
3. Display Fibonacci numbers between m and n

Enter your choice: 3

Enter m (start of range): 100

Enter n (end of range): 500

Fibonacci numbers between 100 and 500 are 144, 233, 377

Press y if you want to continue again?

N

Terminated

TASK 5:

Write a program to simulate a very simple version of Snakes & Ladders. This game contains no snakes and no ladders. Two players compete in this game and the first player that reaches 20 points wins the game. The gameplay is really simple: two players keep on rolling the dice in turn until one of them reaches exact 20 points. If a player rolls a dice and the total points cross 20, this turn is discarded, and the total points don't change.

User is prompted to press any key to roll the dice (explore *getch()* to implement this behavior). You can use *rand()* to implement rolling a dice functionality.

A sample run of this program for single match:

Player 1's turn: 5

Player 1's total: 5

Player 2's turn: 4

Player 2's total: 4

Player 1's turn: 2

Player 1's total: 7

Player 2's turn: 1

Player 2's total: 5

Player 1's turn: 5

Player 1's total: 12

Player 2's turn: 6

Player 2's total: 11

Player 1's turn: 1
Player 1's total: 13
Player 2's turn: 3
Player 2's total: 14

Player 1's turn: 4
Player 1's total: 17
Player 2's turn: 5
Player 2's total: 19

Player 1's turn: 1
Player 1's total: 18
Player 2's turn: 3
Player 2's total: 19 (move not possible)

Player 1's turn: 2
Player 1's total: 20

Player 1 Won!!

Note: There will be a best of 5 matches. The winner will be the player who wins at least 3 matches and if one player wins 3 matches before the fourth or fifth match then no further match will be played.

Good Luck ☺