NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
CS-118 Programming Fundamentals Spring-2021
ASSIGNMENT-03
Section (A, B, C, D, E and F)

Instructions: Dear students we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

i. For each question in your assignment, make a separate cpp file e.g. for question 1, make ROLL-NUM_SECTION_Q#.cpp (20i-0001_A_Q1.cpp) and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in comments.

ii. Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.).

iii. Run and test your program on a machine before submission.

iv. Rename the folder as ROLL-NUM_SECTION (e.g. 20i-0001_A) and compress the folder as a zip file. (e.g. 20i-0001_B.zip). do not submit .rar file.

v. Submit the .zip file on Google Classroom within the deadline.

vi. Submission other than Google classroom (e.g. email etc.) will not be accepted.

vii. The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.

viii. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code. Five Bonus marks will be awarded to well commented/indented code (for all questions).

ix. Total Marks: 100 (10 questions, 10 marks each)

x. If there is a syntax error in code, zero marks will be awarded in that part of assignment.

xi. Your code must be generic.

Deadline: Deadline to submit assignment is **24 April 2021 12:00 AM**. Assignment submitted after the deadline will be marked DIRECT ZERO rather than deducted marks. You are supposed to submit your assignment on GOOGLE CLASSROOM (CLASSROOM TAB not lab). Only ".ZIP" files are acceptable. Other formats should be directly given ZERO. Correct and timely submission of the assignment is the responsibility of every student, hence no relaxation will be given to anyone.

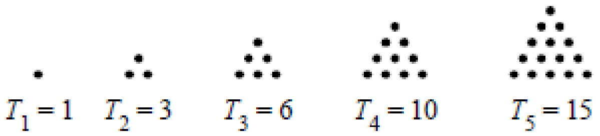
Tip: For timely completion of the assignment, start as early as possible. Plagiarism: Plagiarism is not allowed. **If found plagiarized, you will be awarded zero marks in the assignment.**

Note: Follow the given instruction to the letter, failing to do so will result in a zero.

- For all questions, you can use if else statements, switch statements, do while loops, and while loops. **for loops and nested loops are not allowed.**
- There should be only 10 .cpp files (i.e. 1 .cpp file per question).
- Use of loops (only while **or** do while) is must in every question.
- Example of each question is also given. It’s your responsibility to understand the example thoroughly and attempt the question accordingly. Before doubting or questioning the question statements or examples, put enough time to understand the question and examples yourself.

1. The ancient Greeks classify numbers geometrically. For example, a number was called “triangular” if that number of pebbles could be arranged in a symmetric triangle. The first ten triangular numbers are 0, 1, 3, 6, 10, 15, 21, 28, 36, and 45.

Get an integer n from the user and print **true** if the given integer n is a triangular number, and **false** otherwise. User can enter **any** integer as input.



2. Write a program that finds either **a user given number** is Armstrong or not. If it is Armstrong number, then return **true**, and return **false** if it is not an Armstrong number. **An Armstrong number is an *n*-digit number that is equal to the sum of the *n*th powers of its digits. For example**, 371 is an **Armstrong number** since 3**3 + 7**3 + 1**3 = 371. User can enter any integer of any length.

3. Write a program that will **ask the user** to enter an integer in a range 1-98 (without fraction) and then calculate the sum of the series 1.2+2.3+3.4+4.5+5.6+....... till that integer.

Example:

Input the last integer between 1 to 98 (without fraction) you want to add: 10 (user entered 10)

Series: 1.2 + 2.3 + 3.4 + 4.5 + 5.6 + 6.7 + 7.8 + 8.9 + 9.1 + 10.11 (no need to print the series, only print the sum)

Output: The sum of the series =59.61

don’t accept a number with fraction as input

4. Write a program that asks the user to enter a number of any length and then calculate the product of digits of that number.
Example:
Input a number: 3456
The product of digits of 3456 is: 360

5. Write a program to ask user to input positive integers to process count, maximum, minimum, and average. User will keep entering positive numbers (don't accept -ve numbers other than -1) and when user enters -1, your program will no more ask the user to enter further numbers. Program will then print the following values:

Number of positive integers entered is: ?
Number of odd integers entered is: ?
Number of even integers entered is: ?
The maximum value entered is: ?
The minimum value entered is: ?
The average of all entered numbers is:?

6. Write a program to check whether a **user entered number** is prime or not.
Example:
Input a number to check prime or not: 13 (user entered 13)
output: The entered number is a prime number.
User can enter any number(integer) of any length.

7. Write a program to find the sum of first and last digit of a number entered by the user.
Example:
Input any number: 12345
The first digit of 12345 is: 1
The last digit of 12345 is: 5
The sum of first and last digit of 12345 is: 6
User can enter any number(integer) of any length.

8. Write a program that will ask the user for number of terms N and base b, and then print the first N numbers for a specific base.
Example:
Print the first N numbers for a specific base:
The number 11 in base 10 = 1*(10^1)+1*(10^0)=11
Similarly the number 11 in base 7 = 1*(7^1)+1*(7^0)=8

Input:

Input the number of terms N: 15

Input the base b: 9

Output:

The first 15 numbers in base 9 are:

1 2 3 4 5 6 7 8 10 11 12 13 14 15 16

9. Write a program that asks the user to enter a number n and then displays the first n terms of Fibonacci series.

Example:

Input number of terms to display: 10

Here is the Fibonacci series upto to 10 terms:

0 1 1 2 3 5 8 13 21 34

10. Write a program in C++ to find the Greatest Common Divisor (GCD) of two integer numbers entered by the user.

Sample Output:

Input the first number: 25

Input the second number: 15

The Greatest Common Divisor is: 5