Programming Fundamentals (CS 1002)

FALL 2021 ASSIGNMENT # 4

Due Date: Monday, November 29, 2021 (11:59 pm)

Instructions

Please follow the following submission instructions. Failure to submit according to the above format would result in deduction of 10% marks. Submissions on the email will not be accepted.

- Combine all your work (solution folder) in one .zip file. Use proper naming convention for your submission file. Name the .zip file as **SECTION_ROLL-NUM_04.zip** (**e.g. A_21i0412_04.zip**). Your zip file should only contain **.cpp** files, each file should correspond to its question/problem number. Submit .zip file on Google Classroom within the deadline.

Plagiarism: Plagiarism cases will be strictly dealt with. If found plagiarized, both the involved parties will be awarded zero marks in this assignment, all of the remaining assignments, or even an **F grade** in the course. Copying from the internet is the easiest way to get caught!

Deadline: The deadline to submit the assignment is **Monday, November 29, 2021 (11:59 pm)**. Late submission with marks deduction will be accepted. Correct and timely submission of the assignment is the rresponsibility of every student; hence no relaxation will be given to anyone.

Modular Programming: Your code for each question should be modular. Use functions to decompose your problem into smaller subproblems. This is also mentioned in rubrics.

Rubrics: Make sure that you are aware of rubrics for each question given after that question.

Note: Start early so that you can finish it on time

Problem 1: Pattern Making using loops. (25 Marks):

Make following patterns using loops. Do not hard code anything where there is a repeating pattern.

a.	|......|....................| |......|.......***....*.....| |......|......**....*.*.*...| |......|.....**.....*...*...| |......|......**............| |......|........***.........| |......|....................|
b.	1 232 34543 4567654 567898765 4567654 34543 232 1
c.	1234567654321 12345654321 123454321 1234321 12321 121 1
d.	* ** **** *********** *** *** ** ** * *

e.	-.* -.*-.* -.*-.*-.* -.*-.*-.*-.* -.*-.*-.*-.*-.*-.*

Rubrik (25 marks)
Correct spacing (2 marks for each part)
printing using functions (2 marks for each part)
Correct output (1 mark for each part)

Problem 2: [Repeat from previous but using loops now] Write a C++ program that takes up to 10-digit integer input from user (can be 1 digit, 2 digit, and so on..); passes this to a function which reverses the digits. Finally, the reversed number should be displayed in the main function. For example: when user enters 10-digit like 1234567890 your function will reverse it to 987654321. Be careful for big integer values. [use functions, decision control]

Rubrik (5 marks)
Function to reverse number (1 marks)
Input validation and proper use of data types (1 mark)
Proper use of loop to reverse the number (3 marks)

Problem 3: Write a C++ program which asks user to enter two integer values and user option to find HCF, LCM.

Rubrik (5 marks)
Good program structure. Making functions and using switch (1 marks)
Logic and proper use of loop for each operation (2 x 2 = 4 marks)

Problem 4: The Math(cmath) library contains a method that allows you to calculate the trigonometric function cos(x). You can execute this method by writing cos(x) for some expression x of type double. Using properties of a Maclaurin series or Taylor series, you can approximate the function cos(x) using the following formula:

$$\cos(x) \approx \sum_{n=0}^{a} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \;\ldots.. \frac{x^{2a}}{2a!}$$

Write your own approximation of cos(x) without using **any library** methods(such as pow() etc). Your program should take x (in radians) and a as input. Do input validation to consider that 360^o = 2∏ which is the maximum value of x. You should make two functions calcTerm() and sumTerms() to calculate the value of cos(x). calcTerm() will compute nth term in the sequence. It should take x and term number as arguments and return value of the term. sumTerms() takes single argument term value and should be used to calculate the sum of all terms. Finally your program will have a function testFunction(), you will call it from main to verify whether your function is working correctly or not.

Rubrik (20 marks)
3 marks for each function
-1 for not showing results in main function, for each task
-1 for not passing a value to function in parameter, for each task
Verification of your implementation (5 marks)

Problem 5: A perfect number is an integer that is equal to the sum of its factors. For example, 6 is a perfect number as 6 = 3+2+1. Write down a program that takes an integer x as an input and prints all perfect numbers from 1 to x.

Rubrik (5 marks)
input number(1mark)
printing values using loops (2)
function for finding factor (2 marks)

😊 Happy coding: Start you coding now 😊