NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD

Programming Fundamentals

Fall 2022 ASSIGNMENT # 4

Due Date ≈ Wednesday, November 8th, 2022 (11:59 pm)

Instructions

Submission: Combine all your work 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_22i1412_04.zip**). Your zip file should not contain any folders or subfolders. It should only contain .cpp files for each question, e.g. Q1.cpp, Q2.cpp, Q3.cpp, Q4.cpp, Q5.cpp OR if additional files are asked they will be mentioned with each question. Submit .zip file on Google Classroom within the deadline. Failure to submit according to the above format would result in **25% marks deduction**. Submissions on the email will not be accepted.

Plagiarism: Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in this assignment, all 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 8th **November 2022 at 11:59 PM**. Late submission with marks deduction will be accepted according to the course policy shared earlier. Correct and timely submission of the assignment is the responsibility of every student; hence no relaxation will be given to anyone.

Bonus: In case you implement any additional feature which you think is worth of bonus, make it prominent so that we can see it at runtime.

Note:

- *Each question will be graded based on your effort, additional marks will be awarded for using good programming practices, well-written, good design and properly commented.*
- All programs must be generic.
- Your code should be modular. Use the divide and conquer approach.
- Follow the given instructions to the letter, failing to do so will result in a zero.

Question 1: Design an accurate clock that displays time on the console screen in the following format

X Hours: Y Minutes: Z Seconds

The clock will be 24-hour clock, i.e. the hours will go up to 24, the screen must also reset after every 10 seconds to show updated time. The clock will run infinitely until the program is terminated manually. **[10 Marks]**

Hint: you will need to use <unistd.h> header file if you are working on Linux and <windows.h> header file if you are working on windows, the functions required are system() and Sleep().

Question 2: Pattern printing

a) Draw a circle pattern as shown below of size n (diameter) **[10 Marks]**

For diameter = 9 (radius = 4.5)	For diameter = 11	For diameter = 13
...***... ..*****.. .*******. ********* ********* ********* .*******. ..*****.. ...***...	....***.... ...*****... ..*******.. .*********. *********** *********** *********** .*********. ..*******.. ...*****... ....***....	.....***..... ....*****.... ...*******... ..*********.. .***********. ************* ************* ************* .***********. ..*********.. ...*******... ....*****.... .....***.....

b) Draw the following pattern having alternating dot (.) and underscore (_) characters. It is important to understand that the alternating sequence is across the lines as well, i.e., if a line starts with a dot, then the next line will start with an underscore and vice versa. The input from the user is the number of lines until the centerline. **[10 Marks]**

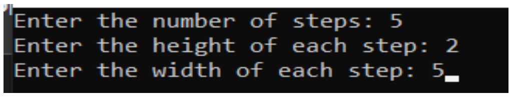
For n = 5

```
*_._.
**._.
***_.
****.
*****
****_
***._
**_._
*._._
```

Bonus: To get 10 bonus marks, draw the above pattern using only one loop.

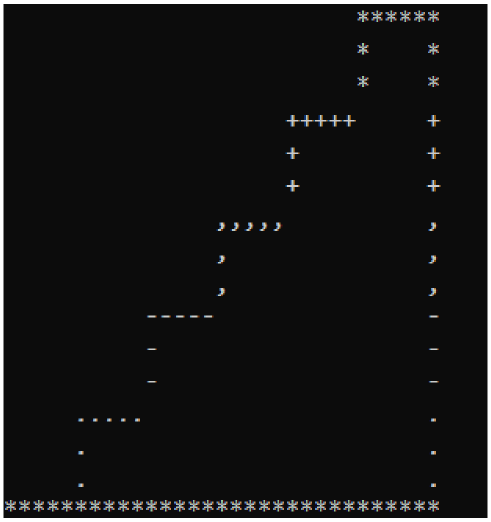
Question 3: You are required to simulate an escalator using loops, functions and iomanip. The program will take 3 inputs at the start of program and then draw the escalator pattern based on those inputs, the loop will run infinitely in which each step will move forward until it reaches top of escalator, after reaching the top it will become the first step again. You will need to use concepts learned in question 1 to implement question 6. A video containing sample inputs and resulting outputs has been attached with the assignment. Use it to solve the question. Each step in the escalator must be a different symbol to show upward movement of steps. (The escalator is slowed down in the video for your understanding, you can make it faster). **[20 Marks]**

Inputs:



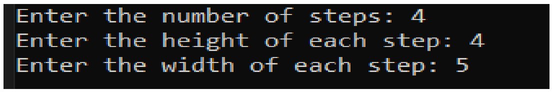
```
Enter the number of steps: 5
Enter the height of each step: 2
Enter the width of each step: 5
```

Output:



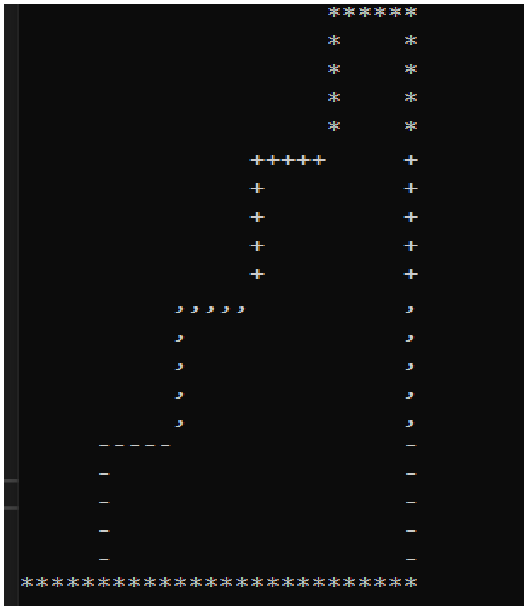
```
                         ******
                         *    *
                         *    *
                    +++++     +
                    +         +
                    +         +
               ,,,,,          ,
               ,              ,
               ,              ,
          -----               -
          -                   -
          -                   -
     .....                    .
     .                        .
     .                        .
*******************************
```

Input:



```
Enter the number of steps: 4
Enter the height of each step: 4
Enter the width of each step: 5
```

Output:



```
                    ******
                    *    *
                    *    *
                    *    *
                    *    *
               +++++     +
               +         +
               +         +
               +         +
               +         +
          ,,,,,          ,
          ,              ,
          ,              ,
          ,              ,
          ,              ,
     -----               -
     -                   -
     -                   -
     -                   -
     -                   -
**************************
```

As you can see from the image, first input decides total steps, 2nd input decides total symbols between each step (step height) and third input decides the total symbols in each step (step width)

Hint: The ASCII value of each consecutive step in the image has a difference of one

Question 4: In this question you are required to design a game show for a single contestant. The contestant will be able to answer questions in 4 categories: Sciences, History, Sports and Pop Culture. The probability of them answering a question in any category must be randomized between 50%-100%. Which question the contestant answers at each round will be also randomized between the 4 categories mentioned above. Whenever a question is correctly answered, their prize money will increase and the probability of that category falls by 10%. The user can stay in the game until their probability of answer a question is above 20%. The game ends on its own if the contestant has been asked 15 questions. At the end of the game the program should display the prize money the contestant has won and how many questions they have answered at each round and what category were questions they answered at each round. **[15 Marks]**

The game must be designed using functions while returning the winning amount to main function and displaying it in main function.

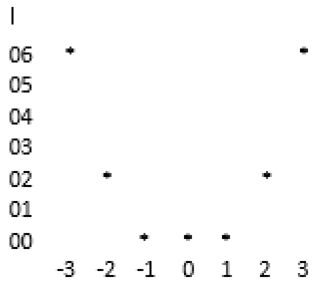
The prize scheme follows

Questions Answered	Prize Money Increase
1-3	100 per question
4-6	1000 per question
7-10	10,000 per question
10-15	15,000 per question

You will be scored on the amount of information you display during each round and the aesthetic look of your outputs, and in order to randomize the game you need to use srand() and time() functions from ctime library.

Question 5: Plotting function: Write a program to plot a graph of function f(x) = x^n - x^{n-1}

Your program should take a maximum absolute value of x as well as a positive integer n as input. You will plot a graph for the range [-x, x]. You should label the y-axis according to the maximum and minimum value of x.
Sample Output: For x = 3 and n = 2, you should have the following output. **[20 Marks]**



```
|
06   *                       *
05
04
03
02       *               *
01
00           *   *   *
    -3  -2  -1   0   1   2   3
```

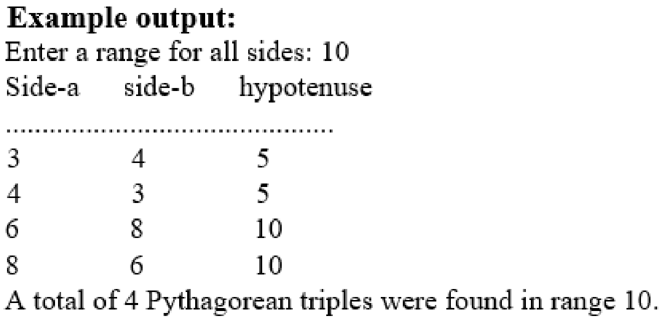
Question 6: Given the first value, generate the next seven terms of the sequences like 1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211. **[10 Marks]**

Question 7: In this problem, you are required to use for loop only to implement a calculator using ++ (increment) and -- (decrement) operators only. The operations that your calculator shall perform are limited, i.e. **addition, subtraction, multiplication, division, and remainder**. Your program shall ask the user about the two input values and an operator, and then use the switch-case structure to run the required operation (addition, subtraction, multiplication, or division). Your program shall run correctly for positive input values only. **[25 Marks]**

First Input: 10
Second Input: 12
Operation: Addition
Answer: 22

Note: Any operation that uses arithmetic operators other than ++ or -- will be awarded zero marks.

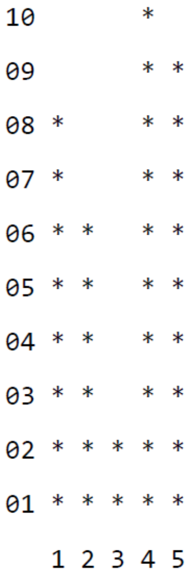
Question 8: Take input from a user and output all possible Pythagorean combinations satisfying the equation $a^2 + b^2 = c^2$ uptill that number. A sample output when input 10 is given is provided below. **[10 Marks]**



Example output:

```
Enter a range for all sides: 10
Side-a     side-b     hypotenuse
.........................................
3          4          5
4          3          5
6          8          10
8          6          10
A total of 4 Pythagorean triples were found in range 10.
```

Question 9: Draw a bar chart of 5 values entered by the user, where the y-axis is along the rows and the x-axis is along the columns of the screen (as shown in the figure below). **[10 Marks]**



```
10       *

09       * *

08 *     * *

07 *     * *

06 * *   * *

05 * *   * *

04 * *   * *

03 * *   * *

02 * * * * *

01 * * * * *

   1 2 3 4 5
```