Assignment 1

Instructions

This assignment is designed to reinforce your understanding of Data Structures and Algorithms. Please read each question carefully and provide clear, well-commented code solutions. Remember to consider efficiency and edge cases in your implementations.

Submission Deadline: Preferably before the upcoming doubt clearing session.

Submission Format: Please maintain the code solutions in your local device for later reference and easy sharing. Maintain the file name format x.y.cpp, where x is the assignment number and y is the problem number.
Ex. File name 2.3.cpp means Assignment 2, 3rd question code.
(.cpp is the extension of a C++ file)

Questions

Question 1:

Problem Statement:

Write a program that takes a student's marks in three subjects: Math, Physics, and Chemistry. Each subject is graded out of 100. Based on these marks, the program should calculate and print the student's final grade as a percentage, along with a pass/fail status and a letter grade.

Constraints & Grading Policy

- **Passing Criteria:** A student passes if they score at least 40 in each subject. If they fail even one subject, their status is "Fail," and their letter grade is "F."
- **Final Percentage:** Calculate the average of the three subjects.
- **Letter Grades:**

- **A:** Final percentage ≥90
- **B:** 80≤ Final percentage <90
- **C:** 70≤ Final percentage <80
- **D:** 60≤ Final percentage <70
- **F:** Final percentage <60 or if the student has failed a subject.

Input Format:

First 3 lines contains the marks of the student in subjects Mathematics, Physics and Chemistry respectively.

Examples:

Input	Output
95 88 92	Final Percentage: 91.67% Status: Pass Letter Grade: A
95 88 35	Final Percentage: 72.67% Status: Fail Letter Grade: F

Question 2:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
*
**
***
****
*****
```

Question 3:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
*
**
***
****
*****
```

Question 4:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
12345
1234
123
12
1
```

Question 5:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
*********
 *******
  *****
   ***
    *
```

Question 6:

Problem Statement:

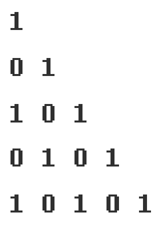
Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
    *
   ***
  *****
 *******
*********
*********
 *******
  *****
   ***
    *
```

Question 7:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

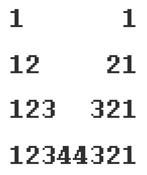


```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

Question 8:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

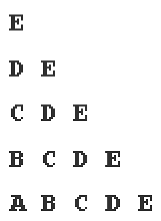


```
1      1
12    21
123  321
12344321
```

Question 9:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.
1 <= x <= 26



```
E
D E
C D E
B C D E
A B C D E
```

Question 10:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
**********
****  ****
***    ***
**      **
*        *
*        *
**      **
***    ***
****  ****
**********
```

Question 11:

Problem Statement:

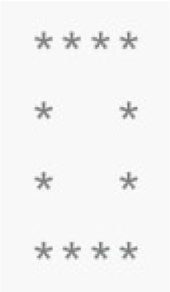
Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
*        *
**      **
***    ***
****  ****
**********
****  ****
***    ***
**      **
*        *
```

Question 12:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

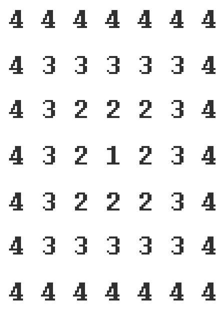


```
****
*  *
*  *
****
```

Question 13:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.



```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4
```

Question 14:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
Input : n = 4
Output :
Solid Rhombus:
   ****
  ****
 ****
****

Hollow Rhombus:
   ****
  *  *
 *  *
****
```

Question 15:

Problem Statement:

Print the following pattern but coming up with a code logic. Keep the program modular, i.e. The pattern should be print for a given input x.

```
Input : 5
Output :
  1
 212
32123
 212
  1

Input : 7
Output :
   1
  212
 32123
4321234
 32123
  212
   1
```

Important Notes:

- Prefer using C++ as the recommended coding language to be used in the bootcamp. Ensure your code is readable and properly commented.
- Please develop the solution logic independently. Relying on AI tools for logic or autocompletion may hinder your ability to learn fundamental syntax and prepare for online assessments with limited GenAI / autocompletion support.
- If you have any questions or need clarification, please reach out on the discord #doubts channel.

Good luck, and happy coding!