Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Subject: CSE-204, Data Structures and Algorithms-I Sessional
Online-1, Spring 2024, Sec-B

**This Lab Test will achieve CO4**

Time: 1.5 hrs **Full Marks: 40**

ID:			**NAME:**		
Marks	**Question-1**	**Question-2**	**Question-3**	**Total**	**Evaluator**
	10	**10**	**20**	**40**	

Question-1

You are a software engineer at a tech company developing a sophisticated markdown text editor. This editor is designed for content creators who rely heavily on precise formatting to produce rich text documents. The editor supports bold and italic formatting using markdown syntax: bold text is marked with ** (double asterisks), and italic text is marked with * (single asterisk). Given the popularity of the editor, users often report issues where the formatted text appears incorrectly due to unbalanced or improperly nested markers.

Enhance the editor by implementing a C program to ensure bold and italic markers are properly balanced and nested in markdown text. This will prevent formatting issues and provide a seamless writing experience. The program should check for correctly balanced and nested bold and italic markers, ignoring other characters.

Requirements:

1. **Input:** A single string containing the text with bold and italic markers.
2. **Output:** A single line: "Valid" if the markers are properly balanced and nested, "Invalid" otherwise.
3. **Constraints:**
    - The input string can be up to 1000 characters long.
    - The text may contain any printable ASCII characters.
    - Bold markers ** and italic markers * must be properly nested and not overlap incorrectly.

Sample Input Output

Input	**Output**
This is **bold** and *italic* text.	Valid
This is **bold* and *italic** text.	Invalid
This is **bold and *italic* text**.	Valid
This is **bold** and *italic text.	Invalid

Marking Criteria	Marks
Stack Implementation	6
Condition Checking	3
Input/ Output	1

Question-2: The Triangle Quest: A Tale of Sticks and Triangles

In the land of Geometry, there exists a set of magical sticks, each with a distinct length. The wise King of Geometry summoned his most trusted mathematician, Arithmo, for an important task. "Arithmo," he said, "you must determine how many valid triangles can be formed using these sticks. A triangle is only valid if it has a positive area. Go forth and solve this puzzle!"

Arithmo knew that for any three sticks to form a triangle, they must satisfy the triangle inequality theorem, which states that **the sum of the lengths of any two sides must be greater than the length of the remaining side**. With this knowledge, Arithmo devised a plan to count all possible valid triangles.

Input starts with a line containing an integer n. The next line contains n integers denoting the lengths of the sticks. You can assume that the lengths are distinct and each length lies in the range [1, 109].

Sample Input Output

Input	Output
5 3 12 5 4 9	No of Valid Triangle: 3
6 1 2 3 4 5 6	No of Valid Triangle: 7
4 100 211 212 121	No of Valid Triangle: 4

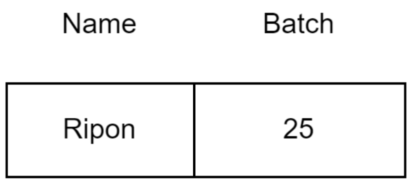
Marking Criteria	Marks
Sorting	3
Binary Search	5
Input/ Output	2

Question 3: Who Gets the Salami?

You were known as the best salami hunter during your days, always receiving a lot of salami from seniors and friends. Now that you've graduated, your juniors are messaging you for salami this Eid. Because of your good relationship with them, you can't ignore these messages. However, as a recent graduate without a job, you don't have much money to give.

You've decided to allocate some money for the salami. You've decided to distribute the salami as follows: 10 taka to your friends, 100 taka to your immediate juniors (CSE-24), 70 taka to CSE-25, and 50 taka to CSE-26. Given your limited resources, you might not be able to give salami to all juniors who message you. To manage this, you plan to implement a linked list-based priority queue that prioritizes message senders as follows: your batchmates will have the highest priority, followed by CSE-24, CSE-25, and CSE-26 consecutively. If you receive multiple messages from the same batch but can only give money to one of them, you'll prioritize the person who messaged you first.

Each node in the linked list will store two pieces of data: 1) the name of the person, and 2) their batch information (see Figure).



An example of individual node

Suppose the current priority queue is: (Ripon, 24) -> (Arif, 25).

Then you open the messenger app and see: (Zara, 24). So the current priority queue will be:

(Ripon, 24) -> (Zara, 24) -> (Arif, 25).

After that, you receive: (Azmain, 23). So the current priority queue will be:

(Azmain, 23) -> (Ripon, 24) -> (Zara, 24) -> (Arif, 25).

Initially, you will input the amount of money you have available for distribution.

Now, your task is to implement four functions:

1. `Push` - This function will be used to insert a value into the queue.
2. `Pop` - This function will remove the person who received the salami.
3. `Peek` - This function will return the person who received the salami without removing them.
4. `Count juniors` - This function will print the count of juniors (see Sample Input/Output for reference).

Sample Input/Output

```
Enter the amount you have in the Salami Fund: 250

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 1
Enter name: Ripon
Enter batch: 24

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 1
Enter name: Arif
Enter batch: 25

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 3
The most eligible salami recipient: (Ripon, 24)

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 1
Enter name: Zara
Enter batch: 24

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 1
Enter name: Azmain
Enter batch: 23
```

Sample Input/Output (cont.)

```
Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 3
The most eligible salami recipient: (Azmain, 23)

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 2
Azmain received 10 taka as salami

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 5
Salami Fund: 240

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 2
Ripon received 100 taka as salami

Options:
1. Push
2. Pop
3. Peek
4. Count juniors
5. Current Salami Fund
6. Exit
Select your option: 4

The number of Juniors and Friends who received salami are:
CSE 23 ->1
CSE 24 ->1
CSE 25 ->0
CSE 26 ->0
```

Marking Criteria	Marks	Marking Criteria	Marks
Push()	10	Peek()	02
Pop()	03	ConuntJuniors()	04
CurrentSalamiFund()	01		