Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	REC_2028_OOPS using Java_Week 9_MCQ
Course	2024_28_III_OOPS Using Java Lab

IP Address 2409:... Tab Switches -- OS Used Windows Browser Used Ch...

Test Duration 00:... Test Start Time N... Test Submit Time N Resume Count 1

Summary Sections Filters

1 MCQ (15)

Question No: 1

Multi Choice Type Question

What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(20);
        list.add(30);
        list.remove(1);
        System.out.println(list);
    }
}
```

○ [10, 30]

○ [20]

Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	2028_REC_OOPS using Java_Week 9_Q1
Course	2024_28_III_OOPS Using Java Lab

IP Address 115.24...	Tab Switches --	OS Used Windows	Browser Used Fir...
Test Duration 00:...	Test Start Time O...	Test Submit Time C	Resume Count 2

Summary Sections

Filters

1 Coding (1)

Question No: 1

Single File Programming Question

Problem Statement

Bobby is tasked with processing a sequence of numbers from a monitoring system. He needs to extract a strictly increasing subsequence using an **ArrayList**. The program should dynamically add numbers to the ArrayList only if they are greater than the last number currently stored in the list. Bobby aims to efficiently utilize the dynamic resizing and indexing features of the ArrayList to solve this problem.

Help Bobby implement this solution.

Input format :

The first line of input consists of an integer **N,** representing the number of elements.

The second line consists of **N** space-separated integers, representing the elements.

Output format :

The output prints the list of integers in increasing sequence, ignoring out-of-order elements.

Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	2028_REC_OOPS using Java_Week 9_Q2
Course	2024_28_III_OOPS Using Java Lab

IP Address	115.24...	Tab Switches	--	OS Used	Windows	Browser Used	Fir...
Test Duration	00:...	Test Start Time	O...	Test Submit Time	C	Resume Count	4

Summary Sections

Filters

1 Coding (1)

Question No: 1

Single File Programming Question

Problem Statement

Vikram loves listening to music and wants to create a simple playlist manager using Java Collections. The playlist supports the following operations:

1. "ADD <song>" → Adds the song to the end of the playlist.
2. "REMOVE <song>" → Removes the first occurrence of the song from the playlist. If the song is not found, do nothing.
3. "SHOW" → Displays all songs in the playlist in order. If the playlist is empty, print "EMPTY".
4. "NEXT" → Moves to the next song in the playlist and prints its name. If the playlist is empty, print "EMPTY".

The playlist maintains a "current song" position that starts at the first song when it's added. The NEXT command moves to the next song and prints it, wrapping around to the first song after reaching the last song. When removing songs, the current position adjusts accordingly to maintain proper navigation.

Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	2028_REC_OOPS using Java_Week 9_Q3
Course	2024_28_III_OOPS Using Java Lab

IP Address 115.24...	Tab Switches --	OS Used Windows	Browser Used Fir...
Test Duration 00:...	Test Start Time O...	Test Submit Time N	Resume Count 3

Summary Sections

Filters

1 Coding (1)

Question No: 1

Single File Programming Question

Problem Statement

Assist Pranitha in developing a program that takes an integer N as input, representing the number of names to be read. Then read N names and store them in an **ArrayList.** Finally, input a search string and output the frequency of that string in the list of names.

Note: Some parts of the code are provided as snippets, and you need to complete the remaining sections by writing the necessary code.

Input format :

The first line of input consists of an integer **N,** representing the number of names to be read.

The following **N** lines consist of **N** names, as a string.

The last line consists of a string, representing the name to be searched.

Output format :

The output prints a single integer, representing the frequency of the specified name in the given list.

Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	REC_2028_OOPS using Java_Week 9_PAH
Course	2024_28_III_OOPS Using Java Lab

IP Address 115.24...	Tab Switches --	OS Used Windows	Browser Used Fir...
Test Duration 00:...	Test Start Time O...	Test Submit Time N	Resume Count 4

Summary Sections Filters

1 Coding (3)

Question No: 1

Single File Programming Question

Problem Statement

Aditi is analyzing stock market trends and wants to find the Next Greater Element (NGE) for each stock price in a list. The Next Greater Element for an element x in an array is the first element to the right that is greater than x. If no greater element exists, return -1 for that position.

Your task is to help Aditi by efficiently computing the Next Greater Element for each element in the given array using a **Stack**.

Example:
Input:
6
4 5 2 10 8 6
Output:
5 10 10 -1 -1 -1
Explanation:
For each element:

Result & Analysis

Attempt 1 of 01

Student	raghu nandhan
Email id	241001183@rajalakshmi.edu.in
Test	REC_2028_OOPS using Java_Week 9_CY
Course	2024_28_III_OOPS Using Java Lab

IP Address 115.24...	Tab Switches --	OS Used Windows	Browser Used Fir...
Test Duration 00:...	Test Start Time O...	Test Submit Time N	Resume Count 2

Summary Sections

Filters

1 Coding (4)

Question No: 1

Single File Programming Question

Problem Statement

Raman, a computer science teacher, is responsible for registering students for his programming class. To streamline the registration process, he wants to develop a program that stores students' names and allows him to retrieve a student's name based on their index in the list.

Raman has decided to use an **ArrayList** to store the names of students, as it provides efficient dynamic resizing and indexing.

Write a program that enables Raman to input the names of students and fetch a student's name using the specified index. If the entered index is invalid, the program should return an appropriate message.

Input format :

The first line of input consists of an integer **n,** representing the number of students to register.

The next **n** lines of input consist of the names of each student, one by one.