

# Rajalakshmi Engineering College

Name: Srithan Saravanan

Email: 240701532@rajalakshmi.edu.in

Roll no: 240701532

Phone: 7200352047

Branch: REC

Department: CSE - Section 7

Batch: 2028

Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 9\_Q1

Attempt : 1

Total Mark : 10

Marks Obtained : 10

#### **Section 1 : Coding**

##### **1. 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.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 7  
3 5 9 1 11 7 13  
Output: [3, 5, 9, 11, 13]

### ***Answer***

```
import java.util.ArrayList;
import java.util.Scanner;
class NumberProcessor{
    private ArrayList<Integer> numList;
    public NumberProcessor(ArrayList<Integer> numList){
        this.numList=numList;
    }
    public void processNumbers(){
        ArrayList<Integer> filteredList=new ArrayList<>();
        for(int num : numList){
            if(filteredList.isEmpty() || num>filteredList.get(filteredList.size()-1)){
                filteredList.add(num);
            }
        }
        System.out.println(filteredList);
    }
}
public class Main{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        int number_of_elements=input.nextInt();
        if(number_of_elements<=0){
            return;
        }
    }
}
```

```
ArrayList<Integer> numList=new ArrayList<>();
for(int ctr=0;ctr<number_of_elements;ctr++){
    numList.add(input.nextInt());
}
NumberProcessor processor=new NumberProcessor(numList);
processor.processNumbers();
}
```

**Status :** Correct

**Marks :** 10/10