

# Rajalakshmi Engineering College

Name: Aldrine Linjoe.s  
Email: 240701033@rajalakshmi.edu.in  
Roll no: 240701033  
Phone: 7092049029  
Branch: REC  
Department: CSE - Section 10  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



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

### REC\_2028\_OOPS using Java\_Week 3\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

#### Section 1 : Coding

##### 1. Problem Statement

Alex is a treasure hunter who collects valuable items during their quests. Each item has a specific point value, and Alex wants to maximize their score by strategically removing items one at a time.

The rule is simple: Alex removes the item with the highest point value in each step until no items are left, summing the values of the removed items to calculate the maximum score.

Help Alex to complete his task.

##### ***Input Format***

The first line of input consists of an integer N, representing the size of the array.

The second line of input consists of N space-separated integers, representing the point values of the items.

### **Output Format**

The output prints "Maximum Sum: " followed by the calculated maximum score after removing all items.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 14

7 14 21 28 35 42 49 56 63 70 77 84 91 98

Output: Maximum Sum: 735

### **Answer**

```
// You are using Java
import java.util.*;
class rec{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int s=0;
        int[] a=new int[n];
        for(int i=0;i<n;i++){
            a[i]=sc.nextInt();
            s+=a[i];
        }
        System.out.print("Maximum Sum:"+s);
    }
}
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement:**

Mason is participating in a coding challenge where he must manipulate an integer array. His task is to replace every element in the array with the next

greatest element to its right. The last element of the array remains unchanged, as there is no element to its right.

Your job is to help Mason write a program that performs this transformation and outputs the modified array.

### ***Input Format***

The first line of input contains an integer  $n$  representing the number of elements in the array.

The second line of input contains  $n$  space-separated integers representing the elements of the array.

### ***Output Format***

The output prints the modified array of  $n$  integers, where each element (except the last one) is replaced by the maximum element to its right, and the last element remains unchanged.

Refer to the sample output for formatting specifications.

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

Input: 6

12 3 91 15 12 14

Output: 91 91 15 14 14 14

### ***Answer***

```
// You are using Java
import java.util.*;
class rec{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++){
            a[i]=sc.nextInt();
        }
        int[] b=new int[n];
        int s=1;
```

```

        for(int i=0;i<n-1;i++){
            int m=a[i+1];
            for(int j=i+1;j<n;j++){
                if(a[j]>m){
                    m=a[j];
                }
            }
            b[i]=m;
        }
        b[n-1]=a[n-1];
        for(int i=0;i<n;i++){
            System.out.printf(" %d",b[i]);
        }
    }
}

```

**Status :** Correct

**Marks : 10/10**

### 3. Problem Statement

Nikila is working as an intern in a software firm and is practicing with a matrix where each row represents a set of numerical values. Her task is to identify the row with the highest sum of its elements and remove that row from the matrix. After removing the row with the highest sum, Nikila needs to print the updated matrix.

Your task is to help Nikila in implementing the same. If there are two or more rows that have same the highest sum, the firstly encountered row is deleted.

#### **Input Format**

The first line of the input consists of two space-separated integers, R and C, representing the number of rows and columns in the matrix, respectively.

The following R lines each contain, C space-separated integers representing the matrix elements.

### **Output Format**

The output prints the matrix after removing the row with the highest sum. Each row should be printed on a new line, with elements separated by a space.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 2 2

1 2

3 4

Output: 1 2

### **Answer**

// You are using Java

```
import java.util.*;
```

```
class rec{
```

```
    public static void main(String[] args){
```

```
        Scanner sc=new Scanner(System.in);
```

```
        int n=sc.nextInt();
```

```
        int m=sc.nextInt();
```

```
        int[][] a=new int[n][m];
```

```
        for(int i=0;i<n;i++){
```

```
            for(int j=0;j<m;j++){
```

```
                a[i][j]=sc.nextInt();
```

```
            }
```

```
        }
```

```
        int[] b=new int[n];
```

```
        int s=0;
```

```
        for(int i=0;i<n;i++){
```

```
            for(int j=0;j<m;j++){
```

```
                s+=a[i][j];
```

```
            }
```

```
            b[i]=s;
```

```
            s=0;
```

```
        }
```

```
        int q=b[0];
```

```
        int t=0;
```

```
        for(int i=0;i<n;i++){
```

```

        if(b[i]>q){
            q=b[i];
            t=i;
        }
    }
    int c=0;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(i!=t){
                System.out.print(" "+a[i][j]);
                c++;
            }
        }
        if(c!=0){
            System.out.println();
            c=0;
        }
    }
}

```

**Status :** Correct

**Marks : 10/10**

#### 4. Problem Statement

Robin is a tech-savvy teenager who is diving into programming.

He is working on a project to find special elements in an array called 'leaders.' Leaders are those exceptional elements that are greater than the sum of all the elements to their right.

Assist Robin in writing this program.

Example

Input:

6

16 28 74 19 25 11

Output:

74 25 11

Explanation:

The element 16 is not greater than the sum of elements to its right ( $28 + 74 + 19 + 25 + 11 = 157$ )

The element 28 is not greater than the sum of elements to its right ( $74 + 19 + 25 + 11 = 129$ )

The element 74 is greater than the sum of elements to its right ( $19 + 25 + 11 = 55$ )

The element 19 is not greater than the sum of elements to its right ( $25 + 11 = 36$ )

The element 25 is greater than the sum of elements to its right (11)

The last element 11 is always a leader since there are no elements to its right.

So, the output is {74, 25, 11}.

### ***Input Format***

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

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

### ***Output Format***

The output prints the special elements in the given array, that are greater than the sum of all the elements to their right.

Refer to the sample output for formatting specifications.

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

Input: 5

3 4 2 5 1

Output: 5 1

**Answer**

```
// You are using Java
import java.util.*;
class rec{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++){
            a[i]=sc.nextInt();
        }
        int s=0;
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                s+=a[j];
            }
            if(a[i]>s){
                System.out.print(a[i]+" ");
            }
            s=0;
        }
    }
}
```

**Status : Correct****Marks : 10/10**