

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 3\_Q1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

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

##### **1. Problem Statement**

Rosh is intrigued by numerical patterns. Today, she stumbled upon a puzzle while working with arrays. She wants to compute the sum of the third-largest and second-smallest elements from a list of integers. She seeks your help to implement a program that solves this for her efficiently.

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

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

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

##### ***Output Format***

The output displays a single integer representing the sum of the third-largest and second-smallest elements in the array.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 10  
10 20 30 40 50 60 70 80 90 100  
Output: 100

### **Answer**

```
import java.util.*;  
  
class ArrayPuzzle {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int N = sc.nextInt();  
        int[] arr = new int[N];  
        for (int i = 0; i < N; i++) {  
            arr[i] = sc.nextInt();  
        }  
        TreeSet<Integer> sortedSet = new TreeSet<>();  
        for (int num : arr) {  
            sortedSet.add(num);  
        }  
        List<Integer> sortedList = new ArrayList<>(sortedSet);  
        if (sortedList.size() < 3) {  
            System.out.println("Not enough unique elements");  
        } else {  
            int thirdLargest = sortedList.get(sortedList.size() - 3);  
            int secondSmallest = sortedList.get(1);  
            System.out.println(thirdLargest + secondSmallest);  
        }  
    }  
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 3\_Q2

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

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

##### **1. Problem Statement**

Monica is interested in finding a treasure but the key to opening is to get the sum of the main diagonal elements and secondary diagonal elements.

Write a program to help Monica find the diagonal sum of a square 2D array.

Note: The main diagonal of the array consists of the elements traversing from the top-left corner to the bottom-right corner. The secondary diagonal includes elements from the top-right corner to the bottom-left corner.

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

The first line of input consists of an integer N, representing the number of rows and columns.

The following N lines consist of N space-separated integers, representing the 2D array elements.

### **Output Format**

The first line of output prints "Sum of the main diagonal: " followed by an integer, representing the sum of the main diagonal.

The second line prints "Sum of the secondary diagonal: " followed by an integer, representing the sum of the secondary diagonal.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 3  
1 2 3  
4 5 6  
7 8 9

Output: Sum of the main diagonal: 15  
Sum of the secondary diagonal: 15

### **Answer**

```
import java.util.*;  
  
class DiagonalSum {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int N = sc.nextInt();  
        int[][] matrix = new int[N][N];  
        for (int i = 0; i < N; i++) {  
            for (int j = 0; j < N; j++) {  
                matrix[i][j] = sc.nextInt();  
            }  
        }  
        int mainDiagonalSum = 0;  
        int secondaryDiagonalSum = 0;  
        for (int i = 0; i < N; i++) {  
            mainDiagonalSum += matrix[i][i];  
            secondaryDiagonalSum += matrix[i][N - 1 - i];  
        }  
    }  
}
```

```
        System.out.println("Sum of the main diagonal: " + mainDiagonalSum);
        System.out.println("Sum of the secondary diagonal: " +
secondaryDiagonalSum);
    }
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 3\_Q3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

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

##### **1. Problem Statement**

You are developing a warehouse management system for a shipping company. The system uses an integer array to represent the weights of packages in a specific order. To verify that the weight capacity is not exceeded, the program needs to calculate the sum of the weights of the first and last packages in the list.

##### **Task:**

Write a code to calculate the sum of the weights of the first and last packages in the list. The program should take an integer array as input and return the total weight of the first and last packages.

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

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

The second line of the input is N space-separated integer values.

### **Output Format**

The output is displayed in the following format:

"Sum of the first and last elements: <>Sum<>"

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 5

10 20 30 40 50

Output: Sum of the first and last elements: 60

### **Answer**

```
import java.util.*;  
  
class PackageWeightSum {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int N = sc.nextInt();  
        int[] weights = new int[N];  
        for (int i = 0; i < N; i++) {  
            weights[i] = sc.nextInt();  
        }  
        int sum = weights[0] + weights[N - 1];  
        System.out.println("Sum of the first and last elements: " + sum);  
    }  
}
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 3\_Q4

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

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

##### **1. Problem Statement**

Sesha is developing a weather monitoring system for a region with multiple weather stations. Each weather station collects temperature data hourly and stores it in a 2D array.

Write a program that can add the temperature data from two different weather stations to create a combined temperature record for the region.

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

The first line of input consists of two space-separated integers N and M, representing the number of rows and columns of the matrices, respectively.

The next N lines consist of M space-separated integers, representing the values of the first matrix.

The following N lines consist of M space-separated integers, representing the values of the second matrix.

#### ***Output Format***

The output prints the addition of the two matrices in N rows and M columns, representing the combined temperature record.

Refer to the sample output for formatting specifications.

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

Input: 3 3

1 2 3

4 5 6

7 8 9

1 1 1

2 2 2

3 3 3

Output: 2 3 4

6 7 8

10 11 12

#### ***Answer***

```
import java.util.*;

class TemperatureMatrixAddition {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int M = sc.nextInt();
        int[][] matrix1 = new int[N][M];
        int[][] matrix2 = new int[N][M];
        int[][] result = new int[N][M];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < M; j++) {
                matrix1[i][j] = sc.nextInt();
            }
        }
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < M; j++) {
```

```
        matrix2[i][j] = sc.nextInt();
    }
}
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        result[i][j] = matrix1[i][j] + matrix2[i][j];
    }
}
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        System.out.print(result[i][j] + " ");
    }
    System.out.println();
}
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

### 2028\_REC\_OOPS using Java\_Week 3\_Q5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

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

##### **1. Problem Statement**

Sharon is creating a program that finds the first repeated element in an integer array. The program should efficiently identify the first element that appears more than once in the given array. If no such element is found, it should appropriately display a message.

Help Sharon to complete the program.

##### ***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 array elements.

### ***Output Format***

If a repeated element is found, print the first element that appears more than once.

If no repeated element is found, print "No repeated element found in the array".

Refer to the sample output for formatting specifications.

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

Input: 8  
12 21 13 14 21 36 47 21

Output: 21

### ***Answer***

```
import java.util.*;  
  
class FirstRepeatedElement {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int[] arr = new int[n];  
        Set<Integer> seen = new HashSet<>();  
        int repeated = -1;  
        for (int i = 0; i < n; i++) {  
            arr[i] = sc.nextInt();  
        }  
        for (int num : arr) {  
            if (seen.contains(num)) {  
                repeated = num;  
                break;  
            }  
            seen.add(num);  
        }  
        if (repeated != -1) {  
            System.out.println(repeated);  
        } else {  
            System.out.println("No repeated element found in the array");  
        }  
    }  
}
```

}

**Status : Correct**

241001049

**Marks : 10/10**

241001049

241001049

241001049

241001049

241001049

241001049

241001049

241001049

241001049

241001049

241001049

241001049

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



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

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

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

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

##### **1. Problem Statement**

In a customer loyalty program, reward points are logged in a sorted array as customers make transactions. Occasionally, due to system errors, duplicate entries for the same transaction may appear. To ensure accurate reward calculations, it's crucial to remove these duplicates from the list.

Write a program to process the array of reward points, removing any duplicates while preserving the order of unique entries. The program should then display the cleaned list of unique reward points and the total count of these unique points.

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

The first line of input consists of an integer N, representing the number of reward points.

The second line consists of N space-separated integers, representing the reward points in sorted order.

### **Output Format**

The first line of output prints the cleaned list of unique reward points separated by a space.

The second line of output prints an integer representing the total count of unique reward points.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 3  
100 100 200

Output: 100 200  
2

### **Answer**

```
import java.util.*;  
class Main{  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int[] arr = new int[n];  
        for(int i = 0; i < n; i++)  
            arr[i] = sc.nextInt();  
        LinkedHashSet<Integer> set = new LinkedHashSet<>();  
        for(int x : arr) set.add(x);  
        for(int x : set)  
            System.out.print(x + " ");  
        System.out.println();  
        System.out.println(set.size());  
    }  
}
```

**Status : Correct**

**Marks : 10/10**

## 2. Problem Statement

Eminem is a billiard player who enjoys playing billiards and also likes solving mathematical puzzles. He notices that the billiard balls on the table are arranged in a grid, and he is curious to find the sum of the numbers written on each ball.

Write a program to find the sum of all the numbers written on each ball in the grid.

### ***Input Format***

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

The second line consists of an integer M, representing the number of columns.

The following lines N lines consist of M space-separated integers, representing the numbers written on each ball.

### ***Output Format***

The output prints an integer representing the sum of all the numbers written on each ball.

Refer to the sample output for the formatting specifications.

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

Input: 3

3

1 2 3

4 5 6

7 8 9

Output: 45

### ***Answer***

```
import java.util.*;
public class Main{
    public static void main(String[] args){
```

```
Scanner scanner = new Scanner(System.in);
int N = scanner.nextInt();
int M = scanner.nextInt();
long sum = 0;
for(int i=0; i<N; i++){
    for(int j = 0; j<M;j++){
        sum += scanner.nextInt();
    }
}
System.out.println(sum);
scanner.close();
}
```

Status : Correct

Marks : 10/10

### 3. Problem Statement

Priya is building a system to automate image transformations using matrix operations. To do this, she needs to multiply two matrices representing pixel data and transformation rules.

Help Priya perform matrix multiplication and print the resulting matrix if the operation is valid.

#### *Input Format*

The first line of input consists of two int values, representing the number of rows R1 and columns C1 of the first matrix.

The next  $R_1 \times C_1$  integers represent the elements of the first matrix.

The next line consists of two int values, representing the number of rows R2 and columns C2 of the second matrix.

The next  $R_2 \times C_2$  integers represent the elements of the second matrix.

#### *Output Format*

If matrix multiplication is possible, print R1 lines, each containing C2 space-separated int values representing the resulting matrix.

Otherwise, print "Matrix multiplication not possible".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2 3

1 2 3

4 5 6

3 2

7 8

9 10

11 12

Output: 58 64

139 154

### **Answer**

```
import java.util.Scanner;
class Main{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        int R1= scanner.nextInt();
        int C1= scanner.nextInt();
        int[][] matrix1 = new int[R1][C1];
        for(int i=0; i<R1;i++){
            for(int j = 0; j<C1;j++){
                matrix1[i][j] = scanner.nextInt();
            }
        }
        int R2= scanner.nextInt();
        int C2= scanner.nextInt();
        int[][] matrix2 = new int[R2][C2];
        for(int i =0; i<R2; i++){
            for(int j=0; j<C2;j++){
                matrix2[i][j] = scanner.nextInt();
            }
        }
        if(C1 != R2){
            System.out.println("Matrix multiplication not possible");
        }
    }
}
```

```

        else{
            int[][] resultMatrix = new int[R1][C2];

            for(int i=0; i<R1;i++){
                for(int j=0; j<C2; j++){
                    for(int k=0; k<C1; k++){
                        resultMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
                    }
                }
            }

            for(int i=0;i<R1;i++){
                for(int j=0; j<C2;j++){
                    System.out.print(resultMatrix[i][j] + (j==C2-1?"\n": ""));
                }
                System.out.println();
            }
        }

        scanner.close();
    }
}

```

**Status : Correct**

**Marks : 10/10**

#### 4. Problem Statement

Egath is participating in a coding hackathon, and one of the challenges requires him to work with an array of integers. The task is to remove exactly one element from the array such that the sum of the remaining elements is a prime number.

Help Egath find the first possible prime sum by removing one element or determining if no such prime sum can be achieved.

#### ***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 array elements.

### ***Output Format***

If removing one element results in a prime sum, print the sum.

If no such prime sum can be achieved by removing exactly one element, print "No valid prime sum found".

Refer to the sample output for formatting specifications.

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

Input: 3

1 2 3

Output: 5

### ***Answer***

```
import java.util.Scanner;
class Main{
    static boolean isPrime(long num){
        if(num<=1){
            return false;
        }
        for(long i=2; i*i<=num;i++){
            if(num%i == 0){
                return false;
            }
        }
        return true;
    }
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        int N = scanner.nextInt();
        long[] arr = new long[N];
        long totalSum = 0;
        for(int i=0; i<N;i++){
            arr[i] = scanner.nextLong();
            totalSum += arr[i];
        }
        long firstPrimeSum = -1;
        for(int i=0; i<N;i++){
```

```
long currentSum = totalSum-arr[i];
if(isPrime(currentSum)){
    firstPrimeSum = currentSum;
    break;
}
if(firstPrimeSum != -1){
    System.out.println(firstPrimeSum);
}else{
    System.out.println("No valid prime sum found");
}
scanner.close();
}
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Dharshini A  
Email: 241001049@rajalakshmi.edu.in  
Roll no: 241001049  
Phone: 8056618801  
Branch: REC  
Department: IT - Section 2  
Batch: 2028  
Degree: B.E - IT

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 : 0

#### **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***

-

**Status :** Skipped

**Marks :** 0/10

## 2. 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**

-

**Status :** -

**Marks :** 0/10

### 3. Problem Statement

Emma is a data analyst working with a grid-based system where each cell contains important numerical data. The grid represents spatial data, inventory records, or structured reports that require periodic updates.

Due to system updates and new requirements, Emma needs to modify the grid in the following ways:

She wants to insert either a new row or a new column at a given position. Later, she needs to delete either a row or a column from the modified matrix.

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

The first line contains two integers rows and cols (the dimensions of the matrix).

The next rows lines contain cols space-separated integers representing the initial matrix.

The next line contains two integers insertType and insertIndex:

- insertType = 0 for row insertion, 1 for column insertion.
- insertIndex is the position where the new row/column should be added.

If inserting a row, the next cols integers represent the new row or If inserting a column, the next rows integers represent the new column.

The next line contains two integers deleteType and deleteIndex:

- deleteType = 0 for row deletion, 1 for column deletion.
- deleteIndex is the position to be deleted.

#### ***Output Format***

The first line of output prints the string "After insertion" followed by the modified matrix with the inserted row or column.

Each row of the matrix is printed on a new line with space-separated integers.

The next line prints the string "After deletion" followed by the final matrix after the specified deletion operation.

Each row of the resulting matrix is printed on a new line with space-separated integers.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 3 3

1 2 3

4 5 6

7 8 9

0 1

10 11 12

1 2

Output: After insertion

1 2 3

10 11 12

4 5 6

7 8 9

After deletion

1 2

10 11

4 5

7 8

### **Answer**

-

Status : -

Marks : 0/10

#### **4. Problem Statement:**

Emma, a budding computer vision enthusiast, is working on a challenging

image processing project. She has a square image represented as a 2D matrix of integers. As part of a special filter operation, she needs to rotate the image by 90 degrees clockwise, but there's a twist – she must perform the rotation in-place, using no extra space.

This means Emma has to rotate the matrix without creating a new one. Your task is to help her implement a Java program that takes this square matrix as input and rotates it within the same structure.

Can you help Emma efficiently rotate the image so that her project can move to the next stage?

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

The first line of input contains a single integer  $n$ , representing the number of rows and columns of the square matrix (i.e., the matrix is of size  $n \times n$ ).

The next  $n$  lines each contain  $n$  space-separated integers, representing the elements of each row of the 2D array.

#### ***Output Format***

The first line of output prints "Rotated 2D Array:"

The next  $n$  lines of output print the rotated matrix.

Each line contains  $n$  space-separated integers representing a row of the rotated matrix.

Refer to the sample output for format specification.

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

Input: 3

1 2 3

4 5 6

7 8 9

Output: Rotated 2D Array:

7 4 1

8 5 2

9 6 3

**Answer**

-

**Status :** -

**Marks :** 0/10