

Rajalakshmi Engineering College

Name: Prince Rohith
Email: 240701399@rajalakshmi.edu.in
Roll no: 240701399
Phone: 6369941431
Branch: REC
Department: CSE - Section 5
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:

Imagine you have an array of integer values, and you're tasked with identifying a pair of elements within the array. This pair of elements should have a sum that is the closest to zero when compared to any other pair in the array.

Your goal is to create a program that solves this problem efficiently. The program should accept an array of integers and return the pair of elements whose sum is closest to zero.

Input Format

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

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

Output Format

The output is displayed in the following format:

"Pair with the sum closest to zero: {value} and {value}"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

9 10 -3 -5 -2

Output: Pair with the sum closest to zero: 9 and -5

Answer

```
import java.util.Scanner;

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();
        int pa = 1000;
        int pb = 1000;
        for(int i=0; i<n; i++) {
            for(int j=0; j<n; j++) {
                if(i == j) continue;
                if((Math.abs(arr[i]+arr[j])) < (Math.abs(pa+pb))) {
                    pa = arr[i];
                    pb = arr[j];
                }
            }
        }
        System.out.printf("Pair with the sum closest to zero: %d and %d", pa, pb);
    }
}
```

Status : Correct

Marks : 10/10

2. 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

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[][] arr = new int[n][n];
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                arr[i][j] = sc.nextInt();
        System.out.println("Rotated 2D Array:");
        for(int j=0; j<n; j++) {
            for(int i=n-1; i>=0; i--)
                System.out.print(arr[i][j] + " ");
            System.out.println();
        }
    }
}
```

Status : Correct

Marks : 10/10

3. 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

```
import java.util.Scanner;

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();
        for(int i=0; i<n-1; i++) {
            int l = arr[i+1];
            for(int j=i+1; j<n; j++)
                if(l < arr[j]) l = arr[j];
            System.out.print(l + " ");
        }
        if(!(n == 0))
            System.out.print(arr[n-1]);
    }
}
```

Status : Correct

Marks : 10/10

4. 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

```
import java.util.Scanner;
```

```
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();  
int sum = 0;  
for(int i=0; i<n; i++) sum += arr[i];  
System.out.println("Maximum Sum: " + sum);  
}  
}
```

Status : Correct

Marks : 10/10