IRC_Java_D6_Java 1DArray_CE_COD

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 120 min

Section 1 - coding

Section Summary

- No. of Questions: 5
- Duration: 120 min

Additional Instructions:

None

Q1. **Youngest and Oldest**

The Pan Am 73 flight from Bombay to New York en route Karachi and Frankfurt was hijacked by a few Palestinian terrorists at the Karachi International Airport.

The senior flight purser Neerja Banhot had to wither her fear and start evacuating the passengers on board. She pleaded the hijackers to release the oldest and the youngest person in the aircraft. Heeding to her plea the chief of the hijacker agreed to let go the oldest and the youngest. Given the ages of the passengers find the oldest and the youngest.

Input Format

The first line of input consists of an integer n, corresponding to the number of passengers in the aircraft.
The next line consists of the age of passengers separated by a space.

Output Format

The output prints the youngest and oldest separated by a space.
Print Invalid Input if n or any one of the ages is negative.

Sample Input

```
5
1 3 5 2 4
```

Sample Output

```
1 5
```

Sample Input

```
-6
```

Sample Output

```
Invalid Input
```

Sample Input

```
6
68 -45
```

Sample Output

```
Invalid Input
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Weighing machines in Sunrise Logistics is not working. Raju, the manager of the division wants to calculate the total weight of received goods. Weight is printed in the goods label. Write a suitable code to help Raju.

Input Format

Number of received goods in first line.
Weight of goods in Second line (Space separated).

Output Format

The output prints the total weight.

Sample Input

```
10
1 9 2 8 3 7 4 6 8 6
```

Sample Output

```
54
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. A common problem in statistics is that of generating frequency distribution of the given data. Assuming that the data consists of n positive integers in the range 1 to 25, write a program that prints the number of times each integer occurs in the data.

Input Format

The first line of the input consists of the value of n.
The next n inputs are the array elements.

Output Format

The output prints the frequency of each data.

Sample Input

```
8
10 20 20 10 10 20 5 20
```

Sample Output

```
10 3
20 4
5 1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Given an array **A** consists of **N** number of elements.If the sum of the element is **"even"** print the **sum** of the element.If the sum of the element is **"odd"** print the **product** of the element.

Input Format

The first line of input contains the number of elements **N**
The second line of input represents the elements **A_1, A_2, A_3 . . . . . A_N**

Output Format

Prints the desired result

Sample Input

```
5
1 2 3 4 6
```

Sample Output

```
16
```

Sample Input

```
4
10 20 52 51
```

Sample Output

```
530400
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **Lucy at the Film Festival**
LucarnosFilm Festival is an annual film festival and is also known for being a prestigious platform for art house films. This time at the Lucarnos Film festival there are N movies screened, each of different genre ranging from drama movies to comedy ones and teen movies to horror ones. Lucy is a huge fan of movies and visited the film festival, but she's not sure which movie she should watch.
Each movie can be characterized by two integers Li and Ri, denoting the length and the rating of the corresponding movie. Lucy wants to watch exactly one movie with the maximal value of Li × Ri. If there are several such movies, she would pick a one with the maximal Ri among them. If there is still a tie, she would pick the one with the minimal index among them.
Write a program to help Lucy pick a movie to watch at the film festival.

Input Format

The first line of the input description contains an integer n. Assume that the maximum value for n as 50.
The second line of the input description contains n integers L1, L2, ...,Ln.
The following line contains n integers R1, R2, ...,Rn.

Output Format

Output a single integer i denoting the index of the movie that Lucy should watch in the film festival. Note that you follow 1-based indexing.

Sample Input

```
2
1 2
2 1
```

Sample Output

```
1
```

Sample Input

```
4
2 1 4 1
2 4 1 4
```

Sample Output

```
2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - coding

Q1

Test Case

Input

```
5
1 3 5 2 4
```

Output

```
1 5
```

Weightage - 10

Input

```
-6
```

Output

```
Invalid Input
```

Weightage - 10

Input

```
6
68 -45
```

Output

```
Invalid Input
```

Weightage - 10

Input

```
10
12 45 78 23 56 89 14 25 36 58
```

Output

```
12 89
```

Weightage - 15

Input

```
-88
```

Output

```
Invalid Input
```

Weightage - 15

Input

```
8
8 5 6 -4
```

Output

```
Invalid Input
```

Weightage - 20

Input

```
18
1 2 5 8 6 7 12 45 86 93 87 54 21 36 69 25 58 74
```

Output

```
1 93
```

Weightage - 20

Sample Input

```
5
1 3 5 2 4
```

Sample Output

```
1 5
```

Sample Input

```
-6
```

Sample Output

```
Invalid Input
```

Sample Input

```
6
68 -45
```

Sample Output

```
Invalid Input
```

Solution

```
 import java.io.*;
import java.util.*;
class YoungestAndOldest {
    public static void main(String [] args) {
        int i,n,sum=0,count=0,min=0,max=0;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        if(n<0) {
            System.out.println("Invalid Input");
        }
        else {
            int a[] = new int[n];
            for(i=0;i<n;i++) {
                a[i] = sc.nextInt();
                if(a[i] <0) {
                    System.out.println("Invalid Input");
                    break;
                }
            }
            min = a[0];
            for(i=1;i<n;i++) {
                if(a[i] >0) {
                count++;
                if(a[i] < min) {
                    min = a[i];
                }
                }
            }
            max = a[0];
            for(i=1;i<n;i++) {
                if(a[i] > max) {
                    max = a[i];
                }
            }
        }
        if(count+1 == n) {
            System.out.println(min+" "+max);
        }
        }
    }
```

Q2

Test Case

Input

```
10
1222 1339 476 1215 6024 7188 1962 558 1970 9019
```

Output

```
30973
```

Weightage - 15

Input

```
25
8323 2552 121 4006 1439 8419 4714 5932 9347 7975
```

Output

```
140365
```

Weightage - 15

Input

```
60
997 290 242 766 59 467 418 522 793 424 985 784 4
```

Output

```
28744
```

Weightage - 15

Input

```
41
5713 8117 8687 3137 9417 3123 4279 2944 5757 5186
```

Output

```
201124
```

Weightage - 15

Input

```
200
1407 2551 9381 7624 9985 7805 6038 2585 1390 4909
```

Output

```
972712
```

Weightage - 20

Input

```
200
-1407 2551 -9381 7624 9985 7805 6038 2585 1390 49
```

Output

```
755388
```

Weightage - 20

Sample Input

```
10
1 9 2 8 3 7 4 6 8 6
```

Sample Output

```
54
```

Solution

```
 import java.util.Scanner;
class SumDemo{
   public static void main(String args[]){
       int count;
       int sum;
       sum=0;
      Scanner scan = new Scanner(System.in);
        count = scan.nextInt();
        int array[] = new int[count];
        for (int i = 0; i < count; i++)
        {
            array[i] = scan.nextInt();
        }
        scan.close();
      for( int num : array) {
          sum = sum+num;
      }
      System.out.println(sum);
   }
}
```

Q3

Test Case

Input

```
8
10 20 20 10 10 20 5 20
```

Output

```
10 3
20 4
5 1
```

Weightage - 25

Input

```
5
12 14 16 12 16
```

Output

```
12 2
14 1
16 2
```

Weightage - 25

Input

```
10
2 8 6 2 4 6 8 10 4 4
```

Output

```
2 2
8 2
6 2
4 3
```

Weightage - 25

Input

```
15
2 4 8 6 2 4 6 8 8 8 10 12 10 10 12
```

Output

```
2 2
4 2
8 4
6 2
```

Weightage - 25

Sample Input

```
8
10 20 20 10 10 20 5 20
```

Sample Output

```
10 3
20 4
5 1
```

Solution

```
 import java.util.Arrays;
import java.util.Scanner;
class Test
{
public static void countFreq(int arr[], int n)
{
    boolean visited[] = new boolean[n];

    Arrays.fill(visited, false);
    for (int i = 0; i < n; i++) {
        if (visited[i] == true)
            continue;
        int count = 1;
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                visited[j] = true;
                count++;
            }
        }
        System.out.println(arr[i] + " " + count);
    }
}
public static void main(String []args)
{
    int i,n;
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    int [] arr = new int[n];
    for(i=0;i<n;i++) {
        arr[i] = sc.nextInt();
    }
    countFreq(arr, n);
}
}
```

Q4

Test Case

Input

```
6
10 0 12 34 56 13
```

Output

```
0
```

Weightage - 10

Input

```
3
44 52 4
```

Output

```
100
```

Weightage - 10

Input

```
7
77 99 11 33 22 3 6
```

Output

```
1095791004
```

Weightage - 10

Input

```
10
12 34 56 78 90 20 34 52 40 66
```

Output

```
482
```

Weightage - 10

Input

```
12
44 32 101 122 432 566 766 844 942 10 12 98
```

Output

```
266338304
```

Weightage - 10

Input

```
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
```

Output

```
190
```

Weightage - 10

Input

```
13
0 0 0 1 0 2 0 1 0 2 3 4 5
```

Output

```
18
```

Weightage - 10

Input

```
18
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Output

```
18
```

Weightage - 10

Input

```
30
12 34 55 7 8  9 0 3 2 9 56 43 78 23 91 39 321
```

Output

```
5150
```

Weightage - 20

Sample Input

```
5
1 2 3 4 6
```

Sample Output

```
16
```

Sample Input

```
4
10 20 52 51
```

Sample Output

```
530400
```

Solution

```
import java.util.Scanner;

class Main {
    public static void main(String args[]) {
        int n;
        Scanner in = new Scanner(System.in);
        n = in.nextInt();
        int i;
        int[] array = new int[n];
        for (i = 0; i < n; i++)
            array[i] = in.nextInt();
        int sum = 0, mul = 1;
        for (int num : array) {
            sum = sum + num;
            mul = mul * num;
        }
        i = (sum % 2 == 0) ? sum : mul;
        System.out.println(i);
    }
}
```

Q5

Test Case

Input

```
4
2 1 4 1
2 4 1 4
```

Output

```
2
```

Weightage - 10

Input

```
2
1 2
2 1
```

Output

```
1
```

Weightage - 10

Input

```
10
1 2 5 6 8 7 4 3 6 5
1 5 6 8 6 3 5 10 2 9
```

Output

```
4
```

Weightage - 10

Input

```
16
1 2 5 8 9 6 4 7 1 2 5 6 8 4 7 6
1 2 5 8 6 8 4 2 10 8 9 6 5 4 5 8
```

Output

```
4
```

Weightage - 15

Input

```
26
1 2 5 6 9 8 4 7 5 6 3 2 1 5 4 8 9 5 2 4 5 7
5 8 6 9 7 4 10 2 5 8 10 1 2 5 6 9 8 4 5 2 6
```

Output

```
16
```

Weightage - 15

Input

```
34
1 2 5 8 6 9 7 4 5 6 3 2 5 8 9 6 5 4 7 5 8 9
1 5 4 8 9 6 10 2 5 8 4 6 9 5 1 5 2 6 8 4 8 4
```

Output

```
7
```

Weightage - 20

Input

```
50
5 8 9 6 4 7 2 1 4 5 3 6 8 5 2 3 6 9 7 4 1 2
10 2 5 8 9 6 5 4 7 2 3 5 6 2 10 5 2 3 6 9 4
```

Output

```
39
```

Weightage - 20

Sample Input

```
2
1 2
2 1
```

Sample Output

```
1
```

Sample Input

```
4
2 1 4 1
2 4 1 4
```

Sample Output

```
2
```

Solution

```
import java.io.*;
import java.util.*;
class LucyAtFlimFestival {
    public static void main(String [] args) {
        int i,j,k=0,l=0,n;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int a[] = new int[n];
        int b[] = new int[n];
        for(i=0;i<n;i++) {
            a[i] = sc.nextInt();
        }
        for(i=0;i<n;i++) {
            b[i] = sc.nextInt();
            a[i]=a[i]*b[i];
        }
        j=a[0];
        for(i=1;i<n;i++)
        {
            if(j<a[i]){j=a[i];k=b[i];}
        }
        for(i=n-1;i>=0;i--)
        {
            if((j==a[i])&&(b[i]>=k)){k=b[i];l=i+1;}
        }
        System.out.println(l);

    }
}
```