IRC_JAVA_COD_ARRAYS1D

Test Summary

- No. of Sections: 1
- No. of Questions: 15
- Total Duration: 180 min

Section 1 - Coding

Section Summary

- No. of Questions: 15
- Duration: 180 min

Additional Instructions:

None

Q1. 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

Q2. Given an array of elements. Find two elements in the array such that their sum is equal to the given element K?

Input Format

The first line of the input consists of the value of n.
The next input is the array elements.
The last input is the sum.

Output Format

The output prints whether the array has a pair of elements with the given sum.
Refer sample output for formatting specifications.

Sample Input

```
6
1 4 45 6 10 -8
16
```

Sample Output

```
Array has two elements with given sum 16
```

Sample Input

```
6
1 4 45 6 10 -8
60
```

Sample Output

```
Array doesn't have two elements with given sum 60
```

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

Q3. Given an array of numbers. Give an algorithm for finding the first element in the array which is repeated.

Input Format

The first line of the input consists of the value of n.

The next input is the array elements.

Output Format

The output prints the first repeated element in the array.

Sample Input

```
7
10 5 3 5 3 4 6
```

Sample Output

```
The first repeating element is 5
```

Sample Input

```
5
1 5 6 8 7
```

Sample Output

```
There are no repeating elements
```

Sample Input

```
8
5 2 4 6 7 2 4 5
```

Sample Output

```
The first repeating element is 5
```

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

Q4. Given an array A of n elements. Find three elements m, n, and k in the array such that $m^2 + n^2 = k^2$?

Input Format

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

Output Format

The first line of the output prints the sum (square) and the elements that correspond to the sum(squares) separated by a space.
The second line of the output prints the element that corresponds to the above answer.
Else, no such triplet exists.
Refer to the sample output for any formatting specifications.

Sample Input

```
5
3 4 8 6 5
```

Sample Output

```
25 9 16
5.0 3.0 4.0
```

Sample Input

```
5
3 4 8 6 12
```

Sample Output

```
No such triplet exists
```

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

Sample Output

```
2
1 2
2 1
```

```
1
```

Sample Input **Sample Output**

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

```
2
```

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

Q6. **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 **Sample Output**

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

```
1 5
```

Sample Input **Sample Output**

```
-6
```

```
Invalid Input
```

Sample Input **Sample Output**

```
6
68 -45
```

```
Invalid Input
```

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

Q7. **Welcome Party**

New Year is shortly arriving and the students of St. Philip's College of Business are eager to receive the freshers for the coming year. The Welcome party for the freshers is going to be organized in a week's time and in connection to that the College Management has ordered the students to renovate their class room block. The Class room block has N rooms in it numbered from 1 to N. Each room is currently painted in one of the red, blue or green colors. Students are given configuration of colors of their class room block by an array consisting of N values. In this array, color red will be denoted by '1', green by '2' and blue by '3'.
The Management wanted the class room block to be repainted such that each class room has same color. For painting, Students have all the 3 color paints available and mixing any 2 color paints will result into 3rd color paint i.e

- 1 + 2 = 3
- 2 + 3 = 1
- 3 + 1 = 2

For example, if a room is already painted in green color, painting that room red color, will make the color of the room blue.
Also, students have many buckets of paint of each color. Simply put, you can assume that they will not run out of paint. Students are a bit lazy, so they does not want to work much and therefore, has asked you to find the minimum number of rooms they have to repaint (possibly zero) in order to have all the rooms with same color as told by the Management. Can you please help them?

Input Format

First line of input contains an integer N, denoting the number of class rooms in the College's class room black. Assume that the maximum value for N as 50.
Next line of input contains N values, denoting the current color configuration of rooms.

Output Format

Print the minimum number of rooms that need to be painted in order to have all the rooms painted with same color i.e red, blue or green.

Sample Input

```
3
1 2 1
```

Sample Output

```
1
```

Sample Input

```
3
1 1 1
```

Sample Output

```
0
```

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

Q8. **Version Management System**

A version Managementsystem (VMS) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.
In this problem we will consider a simplified model of a development project. Let's suppose that there are N source files in the project. All the source files are distinct and numbered from 1 to N.
A VMS which is used for maintaining the project contains two sequences of source files. The first sequence contains M source files that are ignored by the VMS. If a source file is not in the first sequence, then it's considered to be unignored. The second sequence contains K source files that are tracked by the VMS. If a source file is not in the second sequence, then it's considered to be untracked.
A source file can either be or not be in any of these two sequences. Your task is to calculate two values: the number of source files of the project, that are both tracked and ignored, and the number of source files of the project, that are both untracked and unignored.

Input Format

The first line of the input contains three integers N, M and K denoting the number of source files in the project, the number of ignored source files and the number of tracked source files. Assume that the maximum value for N as 50.
The second line contains M distinct integers denoting the sequence A of ignored source files. The sequence is strictly increasing.
The third line contains K distinct integers denoting the sequence B of tracked source files. The sequence is strictly increasing.

Output Format

Output a single line containing two integers: the number of the source files, that are both tracked and ignored, and the number of the source files, that are both untracked and unignored.

Sample Input

```
7 4 6
1 4 6 7
1 2 3 4 6 7
```

Sample Output

```
4 1
```

Sample Input

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

Sample Output

```
1 1
```

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

Q9. Find largest and smallest number in an array.

Input Format

Input consists of n+1 integer inputs.
First line of the input describes the array size 'n',
Followed by n number of array elements.

Output Format

Output displays the smallest and largest number in the array.

Sample Input

```
5
12 4 2 5 22
```

Sample Output

```
smallest value: 2
largest value: 22
```

Sample Input

Sample Output

```
6
20  30  50  4  71  100
```

```
smallest value: 4
largest value: 100
```

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

Q10. Given an unsorted array of unique integers in the range from 1 to N+1. Find the missing element in the array without sorting the array.

Input Format

Input the size of the array

Input the array elements

Output Format

Print the missing element in an array

Sample Input

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

Sample Output

```
5
```

Sample Input

```
15
4 6 5 7 3 1 2 9 8 10 12 15 14 13 16
```

Sample Output

```
11
```

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

Q11. 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

Q12. Write a program to insert an element at a specified position in the array and find the duplicate values of the array of float values.

Input Format

The first line of the input consists of the value of n.
The next input is the array elements.
The third input is the position.
The fourth input is the element to be inserted.

Output Format

The first line of the output prints the newly formed array separated by a space.
Then print the duplicate elements in consecutive lines.
Note: There is an extra space at the end of the first line of output.

Sample Input

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

Sample Output

```
1.0 8.0 2.0 3.0 4.0 5.0 6.0 1.0 2.0 9.0 10.0
1.0
2.0
```

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

Q13. Write a program to find the first and last occurrence of an element in a sorted array.

Input Format

The first line of the input consists of the value n.
Next input is the array elements.
The last input is the element.

Output Format

The output prints the first and last occurrence of the element separated by a space.

Sample Input

```
9
1 3 5 5 5 5 67 123 125
5
```

Sample Output

```
2 5
```

Sample Input

```
9
1 3 5 5 5 5 7 123 125
7
```

Sample Output

```
6 6
```

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

Q14. Write a program to find all pairs of elements in an array whose sum is equal to the given value. Help Guru to write a program to complete this task.

Input Format

Required Sum in first line.
Number of array elements in the second line.
Array elements in third line separated by space.

Output Format

Number pair with sum as shown in the sample output.

Sample Input

```
30
8
14 -15 9 16 25 45 12 8
```

Sample Output

```
14 + 16 =  30
-15 + 45 =  30
```

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

Q15. 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

Answer Key & Solution

Section 1 - Coding

Q1

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 2
```

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);
}
}
```

Q2

Test Case

Input

```
8
12 35 74 96 20 2 6 8
94
```

Output

```
Array has two elements with given sum 94
```

Weightage - 25

Input

```
8
12 35 74 96 20 2 6 8
99
```

Output

```
Array doesn't have two elements with given sum 99
```

Weightage - 25

Input

```
10
12 23 45 56 78 89 14 25 36 85
81
```

Output

```
Array has two elements with given sum 81
```

Weightage - 25

Input

```
10
12 23 45 56 78 89 14 25 36 85
122
```

Output

```
Array doesn't have two elements with given sum 12
```

Weightage - 25

Sample Input

```
6
1 4 45 6 10 -8
```

Sample Output

```
Array has two elements with given sum 16
```

16

Sample Input

```
6
1 4 45 6 10 -8
60
```

Sample Output

```
Array doesn't have two elements with given sum 60
```

Solution

```
import java.util.*;
class Test {
    static boolean hasArrayTwoCandidates(
        int A[],
        int arr_size, int sum)
    {
        int l, r;
        Arrays.sort(A);
        l = 0;
        r = arr_size - 1;
        while (l < r) {
        if (A[l] + A[r] == sum)
            return true;
        else if (A[l] + A[r] < sum)
            l++;
        else
            r--;
    }
    return false;
}
public static void main(String args[])
{
    int i,n,sum;
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    int [] arr = new int[n];
    for(i=0;i<n;i++) {
        arr[i] = sc.nextInt();
    }
    sum = sc.nextInt();
    if (hasArrayTwoCandidates(arr, n, sum))
        System.out.println("Array has two "
                           + "elements with given sum "+sum);
        else
            System.out.println("Array doesn't have "
                               + "two elements with given sum "+sum);
     }
}
```

Q3

Test Case

Input

```
7
10 5 3 5 3 4 6
```

Output

```
The first repeating element is 5
```

Weightage - 25

Input

Output

```
5
8 4 6 2 3
```

```
There are no repeating elements
```

Weightage - 25

Input

Output

```
8
8 2 4 6 7 2 4 8
```

```
The first repeating element is 8
```

Weightage - 25

Input

Output

```
10
4 2 4 6 8 2 1 4 5 6
```

```
The first repeating element is 4
```

Weightage - 25

Sample Input

Sample Output

```
7
10 5 3 5 3 4 6
```

```
The first repeating element is 5
```

Sample Input

Sample Output

```
5
1 5 6 8 7
```

```
There are no repeating elements
```

Sample Input

Sample Output

```
8
5 2 4 6 7 2 4 5
```

```
The first repeating element is 5
```

Solution

```
import java.util.*;
class Main
{
    static void printFirstRepeating(int arr[])
    {
        int min = -1;
        HashSet<Integer> set = new HashSet<>();
        for (int i=arr.length-1; i>=0; i--)
        {
            if (set.contains(arr[i]))
                min = i;

            else
                set.add(arr[i]);
        }
        if (min != -1)
        System.out.println("The first repeating element is " + arr[min]);
        else
        System.out.println("There are no repeating elements");
```

```
    }
    public static void main (String[] args) throws java.lang.Exception
    {
        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();
        }
        printFirstRepeating(arr);
    }
}
```

Q4

Test Case

Input

```
5
3 4 8 6 5
```

Output

```
25 9 16
5.0 3.0 4.0
```

Weightage - 25

Input

```
5
3 4 8 6 12
```

Output

```
No such triplet exists
```

Weightage - 25

Input

```
8
15 20 42 85 75 96 35 25
```

Output

```
625 225 400
25.0 15.0 20.0
```

Weightage - 25

Input

```
8
12 34 56 78 89 34 67 15
```

Output

```
No such triplet exists
```

Weightage - 25

Sample Input

```
5
3 4 8 6 5
```

Sample Output

```
25 9 16
5.0 3.0 4.0
```

Sample Input

```
5
3 4 8 6 12
```

Sample Output

```
No such triplet exists
```

Solution

```
import java.lang.Math;
import java.util.Scanner;
class Main {
    public static void main (String [] args)
        {
            int i,j,k,n,temp;
            Scanner sc = new Scanner(System.in);
            n = sc.nextInt();
            int [] arr = new int[n];
            int [] sq = new int[n];
            for(i=0;i<n;i++) {
                arr[i] = sc.nextInt();
            }
            for(i=0;i<n;i++) {
                sq[i] = arr[i]*arr[i];
            }
            for(i=0;i<n;i++) {
                for(j=i+1;j<n;j++) {
    if(sq[i]>sq[j]) {
       temp = sq[i];
       sq[i] = sq[j];
       sq[j] = temp;
    }
}
}
int flag =0;
 for ( i = n - 1; i >= 0; i--) {
     j = 0;
     k = i - 1;
    while (j < k) {
        if (sq[i] == sq[j] + sq[k]) {
            System.out.println(sq[i]+" "+sq[j]+" "+sq[k]);
            System.out.println(Math.sqrt(sq[i])+" "+Math.sqrt(sq[j])+" "+Math.sqrt(sq[k]));
            flag = 1;
            break;
        }
         else if (sq[i] > sq[j] + sq[k])
            j += 1;
        else
            k -= 1;
        }
    }
    if(flag == 0)
        System.out.println("No such triplet exists");
    }
}
```

Q5

Test Case

Input

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

Output

```
2
```

Weightage - 10

Input

Output

```
2
1 2
2 1
```

```
1
```

Weightage - 10

Input

Output

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

```
4
```

Weightage - 10

Input

Output

```
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
```

```
4
```

Weightage - 15

Input

Output

```
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
```

```
16
```

Weightage - 15

Input

Output

```
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
```

```
7
```

Weightage - 20

Input

Output

```
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
```

```
39
```

Weightage - 20

Sample Input

Sample Output

```
2
1 2
2 1
```

```
1
```

Sample Input

Sample Output

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

```
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);

    }
}
```

Q6

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

Output

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

```
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);
        }
     }
}
```

Q7

Test Case

Input

```
3
1  2  1
```

Output

```
1
```

Weightage - 10

Input

```
3
1  1  1
```

Output

```
0
```

Weightage - 10

Input

```
8
1  2  1  1  3  2  1  1
```

Output

```
3
```

Weightage - 10

Input

```
16
1  2  3  2  1  3  3  1  3  1  2  2  3  3  3  3
```

Output

```
8
```

Weightage - 15

Input

```
26
1 2 3 1 2 2 2 1 3 2 2 1 2 2 3 2 2 2 2 3 2 2 1
```

Output

```
10
```

Weightage - 15

Input

```
34
1 2 1 1 1 2 1 1 1 1 3 2 1 1 1 1 2 3 3 1 1 1 1
```

Output

```
10
```

Weightage - 20

Input

```
50
1 2 3 1 1 2 3 1 1 1 2 3 1 1 1 1 2 1 3 1 1 1 1
```

Output

```
20
```

Weightage - 20

Sample Input

```
3
1 2 1
```

Sample Output

```
1
```

Sample Input

```
3
1 1 1
```

Sample Output

```
0
```

Solution

```
import java.io.*;
import java.util.*;
class Welcomepaarty {
    public static void main(String [] args) {
        int i,j=0,n,k=0,c=0;
        Scanner sc = new Scanner(System.in);
        n=sc.nextInt();
        int a[] = new int[n];
        for(i=0;i<n;i++) {
            a[i] = sc.nextInt();
        }
        for(i=0;i<n;i++)
        {
            if(a[i]==1) {
                k++;
                }
            else if(a[i]==2) {
                j++;
                }
```

```
        else {
            c++;
            }
    }
    if(k>j)
    {
        if(k>c)
        {
            System.out.println(n-k);
        }
        else {
            System.out.println(n-c);
            }
    }
    else if(j>c) {
        System.out.println(n-j);
        }
    else {
        System.out.println(n-c);
        }

}
```

Q8

Test Case

Input

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

Output

```
1 1
```

Weightage - 10

Input

```
7 4 6
1 4 6 7
1 2 3 4 6 7
```

Output

```
4 1
```

Weightage - 10

Input

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

Output

```
3 1
```

Weightage - 10

Input

```
15 10 5
1 2 5 7 8 9 10 11 12 13
1 2 5 12 13
```

Output

```
5 5
```

Weightage - 15

Input

Output

```
12  6  4
1  2  5  6  8  9
1  2  5  12
```

```
3  5
```

Weightage - 15

Input

Output

```
20  15  12
1  2  3  5  6  8  9  10  11  12  14  15  18  19  20
1  2  3  5  6  8  12  13  15  18  19  20
```

```
11  4
```

Weightage - 20

Input

Output

```
26  13  16
1  2  3  4  5  8  9  10  11  20  21  23  26
1  2  3  4  5  8  9  10  11  15  16  18  20  24  25  26
```

```
11  8
```

Weightage - 20

Sample Input

Sample Output

```
7  4  6
1  4  6  7
1  2  3  4  6  7
```

```
4  1
```

Sample Input

Sample Output

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

```
1  1
```

Solution

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

```
{
    for(j=0;j<m;j++)
    {
        if(a[j]==i)
        {
            break;
        }
        if(j==m-1)
        {
            for(l=0;l<k;l++)
            {
                if(b[l]==i)
                {
                    break;
                }
                if(l==k-1){cc++;}
            }
        }
    }
}
System.out.println(c+" "+cc);
    }
}
```

Q9

Test Case

Input

```
10
23   34   13   64   72   90   10   15   9   27
```

Output

```
smallest value: 9
largest value: 90
```

Weightage - 15

Input

```
6
20   30   50   4   71   100
```

Output

```
smallest value: 4
largest value: 100
```

Weightage - 15

Input

```
16
20   30   50   4   71   100 23   34   13   64   72   90
```

Output

```
smallest value: 1
largest value: 100
```

Weightage - 15

Input

```
4
7 8 9 40
```

Output

```
smallest value: 7
largest value: 40
```

Weightage - 15

Input

Output

```
5
32  6  5  4  3
```

```
smallest value: 3
largest value: 32
```

Weightage - 15

Input

Output

```
9
98  87  65  74  25  254  45  12  41
```

```
smallest value: 12
largest value: 254
```

Weightage - 15

Input

Output

```
5
12  4  2  5  22
```

```
smallest value: 2
largest value: 22
```

Weightage - 10

Sample Input

Sample Output

```
5
12  4  2  5  22
```

```
smallest value: 2
largest value: 22
```

Sample Input

Sample Output

```
6
20   30   50   4   71   100
```

```
smallest value: 4
largest value: 100
```

Solution

```
import java.util.Scanner;
class Large_Small{
    public static void main (String args[])
    {
Scanner scan=new Scanner(System.in);
int min,max;
int n=scan.nextInt();//get input from user for array length
int arr[]=new int[n]; //declaring an array of n elements
//for loop takes input from user
for(int i=0; i<n; i++){
   arr[i]=scan.nextInt();//takes input from user for array
   }
   min=arr[0];//assume first element as smallest value
   max=arr[0];//assume first element as largest value
   for(int i=0; i<n; i++){
     if(min>arr[i]){//loop for find minimum elements
       min=arr[i];
     }

     if(max<arr[i]){
       max=arr[i];  //loop for find maximum elements
     }
   }
   System.out.print("smallest value: "+min);
   System.out.print("\nlargest value: "+max);
```

```
}//display result on the result
}
```

Q10

Test Case

Input

```
6
1  2  4  5  6  7
```

Output

```
3
```

Weightage - 20

Input

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

Output

```
3
```

Weightage - 20

Input

```
20
2  5  6  7  9  8  1  3  11  13  15  17  19  20  12  14  16  18
```

Output

```
4
```

Weightage - 30

Input

```
10
10  6  5  7  1  2  3  4  9  11
```

Output

```
8
```

Weightage - 30

Sample Input

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

Sample Output

```
5
```

Sample Input

```
15
4  6  5  7  3  1  2  9  8  10  12  15  14  13  16
```

Sample Output

```
11
```

Solution

```
import java.util.*;
class Main
{
    public static void main(String[] args) {
    int sum = 0;
    int idx = -1;
    int num;
    Scanner sc=new Scanner(System.in);
```

```
        num=sc.nextInt();
        int arr[]=new int[num];
        for(int i=0;i<num;i++)
        {
            arr[i]=sc.nextInt();
            sum += arr[i];
        }
        int total = (num+2)*(num+1)/ 2;

        System.out.println("" + (total - sum));
        }
    }
```

Q11

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);
   }
}
```

Q12

Test Case

Input

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

Output

```
1.0 8.0 2.0 3.0 4.0 5.0 6.0 1.0 2.0 9.0 10.0
1.0
2.0
```

Weightage - 25

Input

```
12
12 25 44 85 76 95 20 22 44 76 20 99
5
88
```

Output

```
12.0 25.0 44.0 85.0 88.0 76.0 95.0 20.0 22.0 44.(
44.0
76.0
```

Weightage - 25

Input

```
15
123 456 789 987 654 321 123 456 789 147 248 369
10
```

Output

```
123.0 456.0 789.0 987.0 654.0 321.0 123.0 456.0 7
123.0
456.0
```

Weightage - 25

Input

```
20
10 20 30 40 50 60 70 80 90 100 110 120 130 140
20
```

Output

```
10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 100.
180.0
```

Weightage - 25

Sample Input

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

Sample Output

```
1.0 8.0 2.0 3.0 4.0 5.0 6.0 1.0 2.0 9.0 10.0
1.0
2.0
```

Solution

```
import java.io.*;
import java.util.*;
class main {
public static void main(String [] args) {
    int i,j,n,pos,ele;
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    double arr[] = new double[n+1];
    for(i=0;i<n;i++) {
        arr[i] = sc.nextDouble();
    }
    pos = sc.nextInt();
    ele = sc.nextInt();
    n++;
     for (i = n-1; i >= pos; i--) {
    arr[i] = arr[i - 1];
     }
    arr[pos - 1] = ele;
    for(i=0;i<n;i++) {
        System.out.print(arr[i]+" ");
    }
    System.out.println();
    for(i=0;i<n;i++) {
        for(j=i+1;j<n;j++) {
            if(arr[i] == arr[j]) {
                System.out.println(arr[j]);
            }
        }
    }
 }
}
```

Q13

Test Case

Input

```
9
1 3 5 5 5 5 67 123 125
5
```

Output

```
2 5
```

Weightage - 25

Input

Output

```
9
1 3 5 5 5 5 7 123 125
7
```

```
6 6
```

Weightage - 25

Input

Output

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

```
8 9
```

Weightage - 25

Input

Output

```
15
1 2 2 3 4 5 5 6 6 8 123 125 168 178 220
8
```

```
9 9
```

Weightage - 25

Sample Input

Sample Output

```
9
1 3 5 5 5 5 67 123 125
5
```

```
2 5
```

Sample Input

Sample Output

```
9
1 3 5 5 5 5 7 123 125
7
```

```
6 6
```

Solution

```
import java.io.*;
import java.util.*;
class Main {
    public static void findFirstAndLast(int arr[], int x)
{
    int n = arr.length;
    int first = -1, last = -1;
    for (int i = 0; i < n; i++) {
        if (x != arr[i])
            continue;
        if (first == -1)
            first = i;
        last = i;
    }
    if (first != -1) {
        System.out.println(first+" "+last);
    }
    else
        System.out.println("Not Found");
}

public static void main(String[] args)
{
    int i,n,x;
    Scanner sc = new Scanner(System.in);
```

```
        n = sc.nextInt();
        int arr[] = new int[n];
        for(i=0;i<n;i++) {
            arr[i] = sc.nextInt();
        }
        x = sc.nextInt();
        findFirstAndLast(arr, x);
    }
}
```

Q14

Test Case

Input

```
20
8
14 -15 9 16 25 45 12 8
```

Output

```
12 + 8 =   20
```

Weightage - 10

Input

```
50
10
1 2 3 40 5 6 7 8 9 10
```

Output

```
40 + 10 =   50
```

Weightage - 10

Input

```
89
5
88 88 87 1 2
```

Output

```
88 + 1 =   89
88 + 1 =   89
87 + 2 =   89
```

Weightage - 10

Input

```
9
10
1 1 1 1 1 1 1 1 1 8
```

Output

```
1 + 8 =   9
1 + 8 =   9
1 + 8 =   9
```

Weightage - 10

Input

```
5
25
1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2
```

Output

```
1 + 4 =   5
1 + 4 =   5
1 + 4 =   5
1 + 4 =   5
```

Weightage - 15

Input

```
10
50
1 2 3 9 8 1 2 3 9 8 1 2 3 9 8 1 2 3 9 8 1 2
```

Output

```
1 + 9 =   10
1 + 9 =   10
1 + 9 =   10
1 + 9 =   10
```

Weightage - 15

Input

```
5
50
-2 -5 10 7 6 -2 -5 10 7 6 -2 -5 10 7 6 -2 -5 1
```

Output

```
-2 + 7 =  5
-2 + 7 =  5
-2 + 7 =  5
-2 + 7 =  5
```

Weightage - 30

Sample Input

```
30
8
14 -15 9 16 25 45 12 8
```

Sample Output

```
14 + 16 =  30
-15 + 45 =  30
```

Solution

```
import java.util.Scanner;

class PairOfInt {
static void  pairs_value(int inputArray[], int inputNumber)
  {
   for (int i =  0; i < inputArray.length; i++)
     {
        for (int j  = i+1; j < inputArray.length; j++)
          {
            if(inputArray[i]+inputArray[j] == inputNumber)
            {
            System.out.println(inputArray[i]+" + "+inputArray[j]+" =  "+inputNumber);
            }
          }
     }
  }

  public static void  main(String[] args)
  {
    Scanner myObj  = new Scanner(System.in);
    int sum = myObj.nextInt();
    int n = myObj.nextInt();
    int array1[];
     array1 =new int[n];
     for(int i=0;i<n;i++)
         array1[i]=myObj.nextInt();

     pairs_value(array1, sum);

  }
}
```

Q15

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

Output

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

```
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);
    }
}
```