Program Name	Description
BSLab5a.java	Using **BufferedReader**, write a java program that prints a times table based on a user input. Your program should print "Times Table: " to the screen, and wait for an user input after that, which will be used to print the times table.

Output

```
Times table: 2

2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

NOTE: The 2 is just to illustrate the output of the program !!!

Program Name	Description
BSLab5b.java	Using **Scanner**, write a java program that prints the price of a ticket based on an input from user. Your program should print "Please, enter your age: " to the screen, and wait for an input after that, which will be used to check the ticket price

Output

```
Please, enter your age: __
```

If age is less the 10 then

    print *"Ticket cost 1 Euro"*

else if age is less than 45 then

    print "Ticket cost 5 Euro"

else

    print "Ticket cost 2 Euro"

NOTE: Only one message will be displayed at a time !!!

Program Name	Description
BSLab5c.java	Write a program that creates and stores 10 random numbers between 10 and 20 (10 and 20 inclusive) in an array and displays it to the screen. Using a **BufferedReader**, your program should ask the user for a number and it should perform a linear search in the array.

Output

```
Index:  0  1  2  3  4  5  6  7  8  9
List.: 10 12 14 15 12 18 20 48 65 19

Please a number: XX
```

If the element is in the list then

Print "BINGO !!! Number **XX** found at position **YY**"

Else

Print "Number NOT Found in the list"

NOTE: XX is the number entered by the user and YY is the position (index) of the element in the list

Program Name	Description
BSLab5d.java	Write a Java program that takes a numbers from user, generate a random number between 100 and 200 (inclusive) and displays the result of **4 basic math operations** (Addition, Subtraction, Multiplication & Division).

Output

```
Enter a number: XXX

1st Number: XXX
2nd Number: YYY ( Random number )

XXX + YYY = Addition_result
XXX - YYY = Subtraction_result
XXX * YYY = Multiplication_result
XXX / YYY = Division_result
```

Replacing **XXX** & **YYY** with the value entered by the users, and each operation should have its **result shown** after each expression.

Program Name	Description
BSExtra.java	Write a program that creates and stores 10 random numbers between 50 and 75 (50 and 75 inclusive) in an array and displays it to the screen. Using a **Scanner**, your program should ask the user for a number shuffles to perform in the list, the program should also display the result of each shuffle individually. It should also display the array with all shuffles performed at the end.

Output

```
Index:  0  1  2  3  4  5  6  7  8  9
List.: 10 12 14 15 12 18 20 48 65 19

Enter the number of shuffles: _4_

Shuffle 1: 3 -> 6
Shuffle 2: 7 -> 6
Shuffle 3: 9 -> 0
Shuffle 4: 2 -> 8

Index:  0  1  2  3  4  5  6  7  8  9
List.: 19 12 65 20 12 18 48 15 14 10
```

Sample Exam Question

A. **Using full Java code**, write a program that ask users to enter their name, and print it back to the screen in the format "Your name is **X**" (where **X** is the name entered by the user)

CHALLENGE QUESTION

1 Equilibrium Index

An Equilibrium Index of a sequence of numbers is an index where the sum of all elements with a lower index is equal to the sum of all elements with a higher index (the number in the sequence at the index is not included in either sum).

In the sequence $\langle 1, 5, -7, 2, 3, -4, 0 \rangle$ the equilibrium indices are 3 and 6 since $1 + 5 + (-7) = 3 + (-4) + 0$ and $1 + 5 + (-7) + 2 + 3 + (-4) = 0$

Given a sequence of numbers, the task is to find the equilibrium indices of the sequence

Input

First line of input contains a single integer N, $0 < N < 10,000,000$. N is the length of the sequence. The following line contains n integers between $-1,000,000$ and $1,000,000$ inclusive, separated by spaces and followed by a newline character.

Output

The output should be a list of space separated equilibrium indices for the given list sorted from lowest to highest index.

Sample Input 1

```
4
6 2 5 1
```

Sample Output 1

```
1
```

Sample Input 2

```
7
1 5 -7 2 3 -4 0
```

Sample Output 2

```
3 6
```