

Started on	Saturday, 30 December 2023, 5:07 PM
State	Finished
Completed on	Saturday, 30 December 2023, 6:00 PM
Time taken	53 mins 19 secs

Question **1**

Correct

Marked out of 25.00

Sam loves chocolates and starts buying them on the 1st day of the year. Each day of the year, x , is numbered from 1 to Y . On days when x is odd, Sam will buy x chocolates; on days when x is even, Sam will not purchase any chocolates.

Complete the code in the editor so that for each day N_i (where $1 \leq x \leq N \leq Y$) in array `arr`, the number of chocolates Sam purchased (during days 1 through N) is printed on a new line. This is a function-only challenge, so input is handled for you by the locked stub code in the editor.

Input Format

The program takes an array of integers as a parameter.

The locked code in the editor handles reading the following input from `stdin`, assembling it into an array of integers (`arr`), and calling `calculate(arr)`.

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes the i th test case as an integer, N_i (the number of days).

Constraints

$$1 \leq T \leq 2 \times 10^5$$

$$1 \leq N \leq 2 \times 10^6$$

$$1 \leq x \leq N \leq Y$$

Output Format

For each test case, T_i in `arr`, your `calculate` method should print the total number of chocolates Sam purchased by day N_i on a new line.

Sample Input 0

```
3
1
2
3
```

Sample Output 0

```
1
1
4
```

Explanation

Test Case 0: $N = 1$

Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 1: $N = 2$

Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 2: $N = 3$

Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 on a new line.

For example:

Input	Result
3	1
1	1
2	4
3	

Input	Result
10	1296
71	2500
100	1849
86	729
54	400
40	25
9	1521
77	25
9	49
13	2401
98	

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class Choc
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int t=sc.nextInt();
8         for(int i=0;i<t;i++)
9         {
10             int n=sc.nextInt();
11             int c=0;
12             for(int j=1;j<=n;j++)
13             {
14                 if(j%2==1)
15                 {
16                     c+=j;
17                 }
18             }
19             System.out.println(c);
20         }
21     }
22 }
```

	Input	Expected	Got	
✓	3	1	1	✓
	1	1	1	
	2	4	4	
	3			
✓	10	1296	1296	✓
	71	2500	2500	
	100	1849	1849	
	86	729	729	
	54	400	400	
	40	25	25	
	9	1521	1521	
	77	25	25	
	9	49	49	
	13	2401	2401	
	98			

Passed all tests! ✓

Question **2**

Incorrect

Marked out of 25.00

Watson gives Sherlock an array of integers. His challenge is to find an element of the array such that the sum of all elements to the left is equal to the sum of all elements to the right. For instance, given the array **arr** = [5, 6, 8, 11], 8 is between two subarrays that sum to 11. If your starting array is [1], that element satisfies the rule as left and right sum to 0.

You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Complete the code in the editor below. It should return a string, either YES if there is an element meeting the criterion or NO otherwise.

It has the following:

· arr: an array of integers

Input Format

The first line contains **T**, the number of test cases.

The next **T** pairs of lines each represent a test case.

- The first line contains **n**, the number of elements in the array **arr**.
- The second line contains **n** space-separated integers **arr[i]** where $0 \leq i < n$.

Constraints

- $1 \leq T \leq 10$
- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$
- $0 \leq i \leq n$

Output Format

For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

Sample Input 0

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

Sample Output 0

```
NO
YES
```

Explanation 0

For the first test case, no such index exists.

For the second test case, **arr[0] + arr[1] = arr[3]**, therefore index **2** satisfies the given conditions.

Sample Input 1

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

Sample Output 1

```
YES
YES
YES
```

Explanation 1

In the first test case, **arr[2] = 4** is between two subarrays summing to **2**.

In the second case, **arr[0] = 2** is between two subarrays summing to **0**.

In the third case, **arr[2] = 2** is between two subarrays summing to **0**.

For example:

Input	Result
3	YES
5	YES
1 1 4 1 1	YES
4	
2 0 0 0	
4	
0 0 2 0	
2	NO
3	YES
1 2 3	
4	
1 2 3 3	

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 public class Sum{
3     public static void main(String[] args)
4     {
5         Scanner sc=new Scanner(System.in);
6         int t=sc.nextInt();
7         int val=0;
8         for(int i=0;i<t;i++)
9         {
10             int n=sc.nextInt();
11             int a[]=new int[n];
12             for(int j=0;j<n;j++)
13             {
14                 a[j]=sc.nextInt();
15                 //System.out.print(a[j]+" ");
16             }
17             for(int k=0;k<n;k++)
18             {
19                 if(k==0 or k==n-1)
20                 {
21                     val+=0;
22                 }
23                 else{
24                     val+=a[k]
25                 }
26             }
27             //System.out.println();
28         }
29     }
30 }
```

Syntax Error(s)

```
Sum.java:19: error: ')' expected
        if(k==0 or k==n-1)
            ^
Sum.java:19: error: ';' expected
        if(k==0 or k==n-1)
            ^
Sum.java:19: error: variable declaration not allowed here
        if(k==0 or k==n-1)
            ^
Sum.java:19: error: not a statement
        if(k==0 or k==n-1)
            ^
Sum.java:19: error: ';' expected
        if(k==0 or k==n-1)
            ^
Sum.java:25: error: '.class' expected
    }
    ^
Sum.java:23: error: 'else' without 'if'
        else{
        ^

7 errors
```

Question **3**

Correct

Marked out of 25.00

Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

For example, they have $m = 6$ to spend and there are flavors costing $cost = [1, 2, 3, 4, 5, 6]$. The two flavors costing **1** and **5** meet the criteria. Using **1**-based indexing, they are at indices **1** and **4**.

Function Description

Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy.

It has the following:

- m : an integer denoting the amount of money they have to spend
- $cost$: an integer array denoting the cost of each flavor of ice cream

Input Format

The first line contains an integer, t , denoting the number of trips to the ice cream parlor. The next t sets of lines each describe a visit. Each trip is described as follows:

1. The integer m , the amount of money they have pooled.
2. The integer n , the number of flavors offered at the time.
3. n space-separated integers denoting the cost of each flavor: $cost[1], cost[2], \dots, cost[n]$.

Note: The index within the cost array represents the flavor of the ice cream purchased.

Constraints

- $1 \leq t \leq 50$
- $2 \leq m \leq 10^4$
- $2 \leq n \leq 10^4$
- $1 \leq cost[i] \leq 10^4, \forall i \in [1, n]$
- There will always be a unique solution.

Output Format

For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

Sample Input

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

Sample Output

```
1 4
1 2
```

Explanation

Sunny and Johnny make the following two trips to the parlor:

1. The first time, they pool together $m = 4$ dollars. Of the five flavors available that day, flavors **1** and **4** have a total cost of $1 + 3 = 4$.
2. The second time, they pool together $m = 4$ dollars. Of the four flavors available that day, flavors **1** and **2** have a total cost of $2 + 2 = 4$.

For example:

Input	Result
2	1 4
4	1 2
5	
1 4 5 3 2	
4	
4	
2 2 4 3	

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class Money
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int t=sc.nextInt();
8         for(int i=0;i<t;i++)
9         {
10             int m=sc.nextInt();
11             int n=sc.nextInt();
12             int a[]=new int[n];
13             int sum=0,flag=0,i1=0,i2=0;
14             for(int j=0;j<n;j++)
15             {
16                 a[j]=sc.nextInt();
17             }
18             for(int k=0;k<n;k++)
19             {
20                 for(int l=k+1;l<n;l++)
21                 {
22                     sum=a[k]+a[l];
23                     if(sum==m)
24                     {
25                         i1=k+1;
26                         i2=l+1;
27                         if(i1>i2){i2=l+1;i1=k+1;}
28                         flag=1;
29                         break;
30                     }
31                 }
32                 if(flag==1)
33                 {
34                     break;
35                 }
36             }
37             System.out.println(i1+" "+i2);
38         }
39     }
40 }
```

	Input	Expected	Got	
✓	2	1 4	1 4	✓
	4	1 2	1 2	
	5			
	1 4 5 3 2			
	4			
	4			
	2 2 4 3			

Passed all tests! ✓

Question 4

Incorrect

Marked out of 25.00

Numeros the Artist had two lists that were permutations of one another. He was very proud. Unfortunately, while transporting them from one exhibition to another, some numbers were lost out of the first list. Can you find the missing numbers?

As an example, the array with some numbers missing, **arr** = [7, 2, 5, 3, 5, 3]. The original array of numbers **brr** = [7, 2, 5, 4, 6, 3, 5, 3]. The numbers missing are [4, 6].

Notes

- If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.
- You have to print all the missing numbers in ascending order.
- Print each missing number once, even if it is missing multiple times.
- The difference between maximum and minimum number in the second list is less than or equal to **100**.

Complete the code in the editor below. It should return an array of missing numbers.

It has the following:

- **arr**: the array with missing numbers
- **brr**: the original array of numbers

Input Format

There will be four lines of input:

n - the size of the first list, **arr**

The next line contains **n** space-separated integers **arr[i]**

m - the size of the second list, **brr**

The next line contains **m** space-separated integers **brr[i]**

Constraints

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq brr[i] \leq 2 \times 10^4$
- $X_{max} - X_{min} < 101$

Output Format

Output the missing numbers in ascending order.

Sample Input

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

Sample Output

```
204 205 206
```

Explanation

204 is present in both arrays. Its frequency in **arr** is **2**, while its frequency in **brr** is **3**. Similarly, **205** and **206** occur twice in **arr**, but three times in **brr**. The rest of the numbers have the same frequencies in both lists.

For example:

Input	Result
10 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205 206 204	204 205 206

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 public class Missing
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int flag=0,val=0;
8         int n = sc.nextInt();
9         int arr[]=new int[n];
10        for(int i=0;i<n;i++)
11        {
12            arr[i]=sc.nextInt();
13        }
14        int m = sc.nextInt();
15        int brr[]=new int[m];
16        for(int i=0;i<m;i++)
17        {
18            brr[i]=sc.nextInt();
19        }
20        int f1[]=new int[1000];
21        int f2[]=new int[1000];
22        int ind=0;
23
24        for(int i=0;i<
25
26        // for(int i=0;i<m;i++)
27        // {
28        //     for(int j=0;j<n;j++)
29        //     {
30        //         if(brr[i]==arr[j])
31        //         {
32        //             break;
33        //         }
34        //         else{
35        //             val=brr[i];
36        //             flag=1;
37        //             break;
38        //         }
39        //     }
40        //     if(flag==1)
41        //     {
42        //         f1[ind]=val;
43        //         ind++;
44        //     }
45        // }
46        for(int i=0;i<f1.length;i++)
47        {
48            System.out.print(f1[i]+" ");
49        }
50    }
51 }
```

Syntax Error(s)

Missing.java:24: error: illegal start of expression
 for(int i=0;i<
 ^

1 error