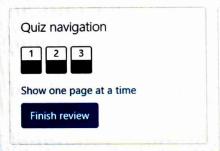
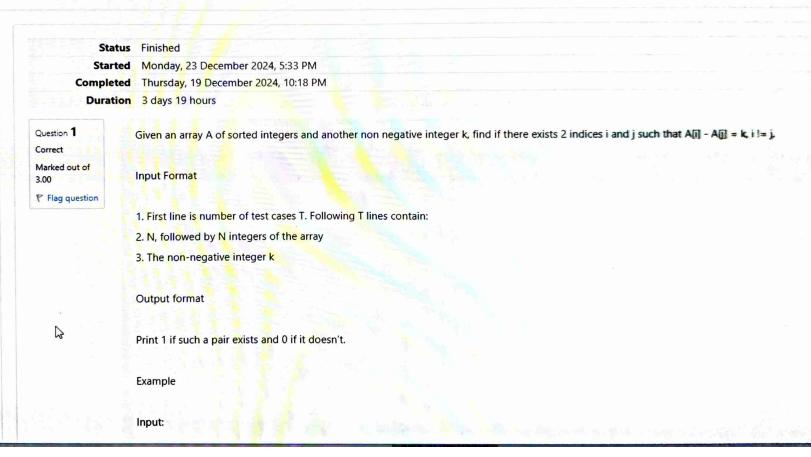
GE23131-Programming Using C-2024





~	Input	Expected	Got	
	1 3 1 3 5 4	1	1	~
~	1 3 1 3 5 99	0	0	~

Passed all tests! ~

Question 2

Correct

Marked out of 5.00

P Flag question

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 Ni (where $1 \le x \le N \le Y$) 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 ith test case as an integer, Ni (the number of days).

Constraints

 $1 \le T \le 2 \times 105$

 $1 \le N \le 2 \times 106$

 $1 \le x \le N \le Y$

Output Format

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

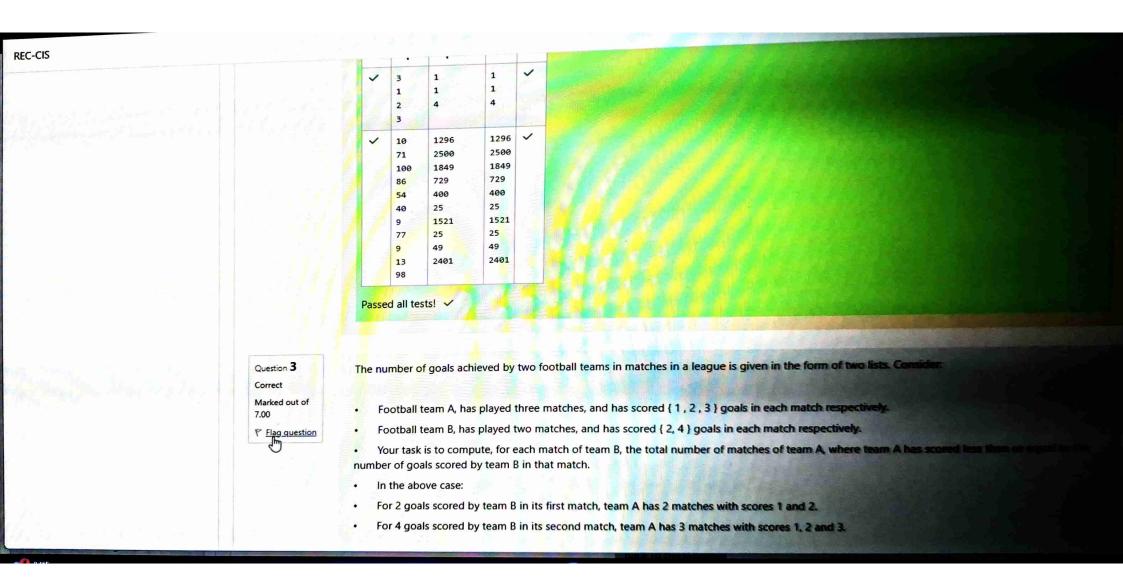
REC-CIS Sample Input 0 3 Sample Output 0 Explanation Test Case 0: N = 1 Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 on a new line. Test Case 1: N = 2 Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 on a new line. 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.

Answer: (penalty regime: 0 %)

```
#include(stdio.h>
int main(){
int t;
            scanf("%d",&t);
           while(t--){
    int n,c=0;
    scanf("%d",&n);
                 for(int i=0;i<=n;i++){
    if(i%2!=0) c=c+i;
10
                 printf("%d\n",c);
11
12
13 }
```

D

	Input	Expected	Got	
~	3	1	1	~
	1	1	1	
	2	4	4	
Ť.	3	Lines		-1.
~	10	1296	1296	~
	71	2500	2500	100



Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of m positive integers, one for each maxes[i] representing the total number of elements nums[j] satisfying nums[j] \leq maxes[i] where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

It has the following:

nums[nums[0],...nums[n-1]]: first array of positive integers
maxes[maxes[0],...maxes[n-1]]: second array of positive integers

Constraints

- 2 ≤ n, m ≤ 105
- $1 \le \text{nums}[j] \le 109$, where $0 \le j < n$.
- 1 ≤ maxes[i] ≤ 109, where 0 ≤ i < m.

Input Format For Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number of elements in nums.

The next n lines each contain an integer describing nums[j] where $0 \le j < n$.

The next line contains an integer m, the number of elements in maxes.

The next m lines each contain an integer describing maxes[i] where $0 \le i < m$.

1

2. For maxes[1] = 5, we have 4 elements in nums (nums[0] = 1, nums[1] = 4, nums[2] = 2, and nums[3] = 4) that are 4 maxes[1].

We are given, n = 5, nums = [2, 10, 5, 4, 8], m = 4, and maxes = [3, 1, 7, 8].

- For maxes[0] = 3, we have 1 element in nums (nums[0] = 2) that is ≤ maxes[0].
- 2. For maxes[1] = 1, there are 0 elements in nums that are ≤ maxes[1].
- 3. For maxes[2] = 7, we have 3 elements in nums (nums[0] = 2, nums[2] = 5, and nums[3] = 4) that are ≤ maxes[2].
- For maxes[3] = 8, we have 4 elements in nums (nums[0] = 2, nums[2] = 5, nums[3] = 4, and nums[4] = 8) that are ≤ maxes[3].

Thus, the function returns the array [1, 0, 3, 4] as the answer.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
  2 v int main(){
         int s1,s2,ans;
         scanf("%d", &s1);
         int ta[s1];
         for(int i=0;i<s1;i++)
 6
         scanf("%d",&ta[i]);
         scanf("%d", &s2);
        int tb[s2];
10
        for(int i=0;i<s2;i++)
11
        scanf("%d",&tb[i]);
        for(int j=0; j<s2; j++)
12
13
14
            ans=0;
            for(int i=0;i<s1;i++){
15 .
                if(tb[j]>=ta[i])
16
17
                ans++;
18
            printf("%d\n",ans);
19
20
21 }
```

5

	Input	Expected	Got		
~	4	2	2	~	
	1	4	4		
	4				
	2				
	4				
	2				
	3				
	5				
_	5	1	1	~	
	2	0	0		
	10	3	3		
	5	4	4		
	4				
	8				
	4				
	3				
	3				
	7		- 1		
	8				