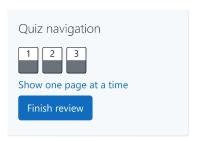
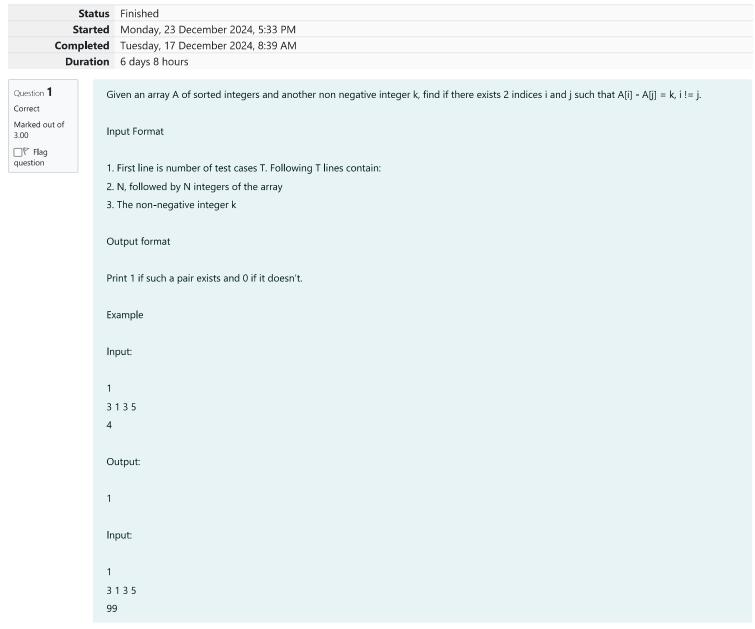
GE23131-Programming Using C-2024





Answer: (penalty regime: 0 %)

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

□ Flag

question

of

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).

REC-CIS

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. Sample Input 0 3 2 3 Sample Output 0 4 Explanation Test Case 0: N = 1Sam 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 = 2Sam 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 = 3Sam 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 %)

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

Question **3**Correct

Marked out of 7.00

□ Flag question The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

- Football team A, has played three matches, and has scored { 1, 2, 3, 3 } goals in each match respectively.
- Football team B, has played two matches, and has scored { 2, 4 } goals in each match respectively.
- Your task is to compute, for each match of team B, the total number of matches of team A, where team A has scored less than or equal to the number of goals scored by team B in that match.
- In the above case:

Passed all tests!

- For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2.
- For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

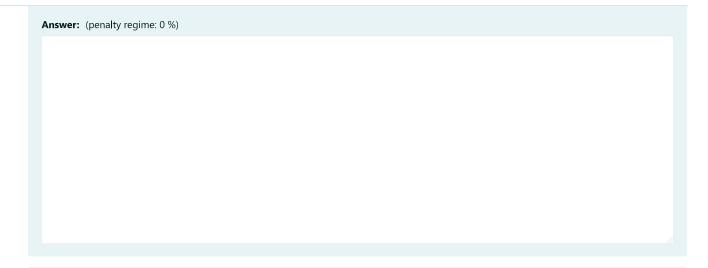
Hence, the answer: {2, 3}.

REC-CIS

```
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 \le n, m \le 105
• 1 \le \text{nums}[j] \le 109, where 0 \le j < n.
   1 \le \text{maxes}[i] \le 109, where 0 \le 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.
Sample Case 0
Sample Input 0
4
4
2
2
3
5
Sample Output 0
2
4
```

REC-CIS

```
We are given n = 4, nums = [1, 4, 2, 4], m = 2, and maxes = [3, 5].
1. For maxes[0] = 3, we have 2 elements in nums (nums[0] = 1 and nums[2] = 2) that are \leq maxes[0].
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 \le maxes[1].
Thus, the function returns the array [2, 4] as the answer.
Sample Case 1
Sample Input 1
5
2
10
5
8
3
7
8
Sample Output 1
1
0
3
Explanation 1
We are given, n = 5, nums = [2, 10, 5, 4, 8], m = 4, and maxes = [3, 1, 7, 8].
1. For maxes[0] = 3, we have 1 element in nums (nums[0] = 2) that is \leq maxes[0].
2. For maxes[1] = 1, there are 0 elements in nums that are \leq 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 \leq maxes[2].
4. 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 \le maxes[3].
```



Inpu	ut Expec	ted Got
4 1 4 2 4 2 3 5	2 4	2 4
5 2 10 5 4 8 4 3 1 7 8	1 0 3 4	1 0 3 4

Passed all tests!

Save the state of the flags

Finish review