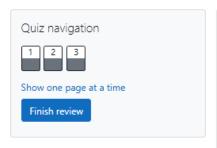
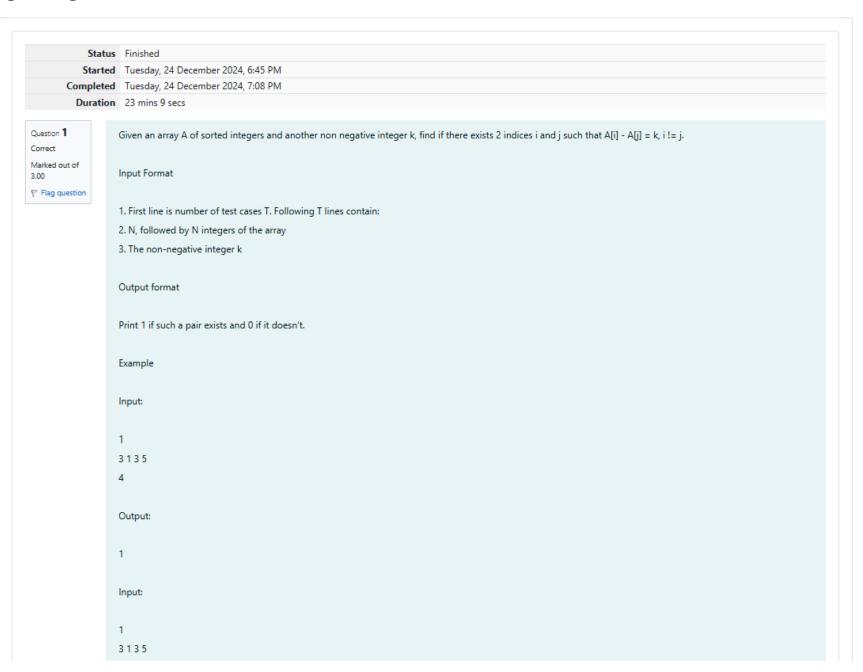
GE23131-Programming Using C-2024





Output:

0

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<math.h>
3 #include<stdlib.h>
5
    int main()
6 + {
7
       int t,n,k,arr[1000],i,j,flag;
8
       scanf("%d",&t);
9
       while(t--)
10
           scanf("%d",&n);
11
12
            for(i=0;i<n;i++)
            scanf("%d",&arr[i]);
13
14
           scanf("%d",&k);
15
           i=0;
16
           j=1;
            flag=0;
17
18
            while(i<n&&j<n)
19
20
               if(abs(arr[i]-arr[j])<k)
21
22
               else if(abs(arr[i]-arr[j])>k)
23
24
               else if(abs(arr[i]-arr[j])==k)
25
                   flag =1;
26
27
                   break;
28
29
30
31
            if(flag==1)
           printf("1\n");
32
33
            else
34
           printf("0\n");
35
36
       return 0;
37
38
```

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

99

Passed all tests! <

Question 2
Correct
Marked out of 5.00

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.

Sample Input 0

3

1

2

3

Sample Output 0

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.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
 2
    int main()
3 √ {
 4
       int testcase =0,N[250],i=0,j=0,count;
 5
       scanf("%d",&testcase);
 6
        for(i=0;i<testcase;i++)</pre>
7
           scanf("%d",&N[i]);
 8
       for(i=0;i<testcase;i++)</pre>
9 ,
10
            count = 0;
11
            for(j=1;j<=N[i];j++)</pre>
12
               if(j%2==1)
13
14
                  count+=j;
15
               else
16
               continue;
17
           printf("%d\n",count);
18
19
20
        return 0;
21 }
```

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 **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 } 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:
- 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}.

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 ≤ nums[j] ≤ 109, where 0 ≤ 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.
Sample Case 0
Sample Input 0
5
Sample Output 0
2
Explanation 0
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 ≤ maxes[1].
Thus, the function returns the array [2, 4] as the answer.
Sample Case 1
Sample Input 1
5
2
10
```

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].
- 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
    int main()
 3 ₹ {
 4
        long maxes[100000],nums[100000],n,m,sum[100000],j,i,count=0;
        scanf("%ld",&n);
 5
 6
       for(i=0;i<n;i++)
7
        scanf("%ld",&nums[i]);
 8
       scanf("%ld",&m);
9
       for(j=0;j<m;j++)
10
          scanf("%ld",&maxes[j]);
11
        for(i=0;i<m;i++)
12
13
            for(j=0;j<n;j++)</pre>
14 ,
15
                if(nums[j]<=maxes[i])</pre>
16
               count = count+1;
17
            sum[i]=count;
18
19
            count=0;
```

```
for(i=0;i<m;i++)
printf("%ld\n",sum[i]);
return 0;
}
```

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

Passed all tests! <

Finish review