

<b>Status</b>	Finished
<b>Started</b>	Thursday, 13 November 2025, 12:14 PM
<b>Completed</b>	Thursday, 13 November 2025, 12:51 PM
<b>Duration</b>	36 mins 49 secs

**Question 1**

Correct

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that  $A[i] - A[j] = k$ ,  $i \neq j$ .

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Example

Input:

1  
3 1 3 5  
4

Output:

1

Input:

1  
3 1 3 5  
99

Output:

0

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int T;
5     scanf("%d",&T);
6     while(T--)
7     {
8         int N;
```

```

9  ... ,  

10 int A[N];  

11 for(int i=0;i<N;i++)  

12 scanf("%d",&A[i]);  

13 int k;  

14 scanf("%d",&k);  

15 int i=0,j=1,found=0;  

16 while(i<N && j<N)  

17 {  

18     int diff=A[j]-A[i];  

19     if(i!=j &&diff==k)  

20     {  

21         found=1;  

22         break;  

23     }  

24     else if(diff<k)  

25         j++;  

26     else  

27         i++;  

28 }  

29 printf("%d\n",found);  

30 }  

31 }  

32 return 0;  

33 }  

34 }
```



	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

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.

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.

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 void calculate(int arr[],int T){
3     for(int i=0;i<T;i++){
4         long long N=arr[i];
5         long long k=(N+1)/2;
6         printf("%lld\n",k*k);
7     }
8 }
9
10 int main()
11 {
12     int T;
13     scanf("%d",&T);
14     int arr[T];
15     for(int i=0;i<T;i++){
16         scanf("%d",&arr[i]);
17     }
18     calculate(arr,T);
19     return 0;
20 }
21
22

```



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

	Input	Expected	Got	
✓	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 3**

Correct

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  $\text{nums}[j] \leq \text{maxes}[i]$  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

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

#### 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  $0 \leq j < n$ .

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  $0 \leq i < m$ .

#### Sample Case 0

### Sample Input 0

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

### Sample Output 0

```
2  
4
```

### Explanation 0

We are given  $n = 4$ ,  $\text{nums} = [1, 4, 2, 4]$ ,  $m = 2$ , and  $\text{maxes} = [3, 5]$ .

1. For  $\text{maxes}[0] = 3$ , we have 2 elements in  $\text{nums}$  ( $\text{nums}[0] = 1$  and  $\text{nums}[2] = 2$ ) that are  $\leq \text{maxes}[0]$ .
2. For  $\text{maxes}[1] = 5$ , we have 4 elements in  $\text{nums}$  ( $\text{nums}[0] = 1$ ,  $\text{nums}[1] = 4$ ,  $\text{nums}[2] = 2$ , and  $\text{nums}[3] = 4$ ) that are  $\leq \text{maxes}[1]$ .

Thus, the function returns the array  $[2, 4]$  as the answer.

### Sample Case 1

#### Sample Input 1

```
5  
2  
10  
5  
4  
8  
4  
3  
1  
7  
8
```

#### Sample Output 1

```
1
```

```

0
3
4

```

### Explanation 1

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

1. For  $\text{maxes}[0] = 3$ , we have 1 element in  $\text{nums}$  ( $\text{nums}[0] = 2$ ) that is  $\leq \text{maxes}[0]$ .
2. For  $\text{maxes}[1] = 1$ , there are 0 elements in  $\text{nums}$  that are  $\leq \text{maxes}[1]$ .
3. For  $\text{maxes}[2] = 7$ , we have 3 elements in  $\text{nums}$  ( $\text{nums}[0] = 2$ ,  $\text{nums}[2] = 5$ , and  $\text{nums}[3] = 4$ ) that are  $\leq \text{maxes}[2]$ .
4. For  $\text{maxes}[3] = 8$ , we have 4 elements in  $\text{nums}$  ( $\text{nums}[0] = 2$ ,  $\text{nums}[2] = 5$ ,  $\text{nums}[3] = 4$ , and  $\text{nums}[4] = 8$ ) that are  $\leq \text{maxes}[3]$ .

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

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 int compare(const void*a,const void*b){
4     return (*(int *)a-*(int*)b);
5 }
6 int upper_bound(int arr[],int n,int key){
7     int low=0,hight=n;
8     while(low<hight){
9         int mid=(low+hight)/2;
10        if(arr[mid]<=key)
11            low=mid+1;
12        else
13            hight=mid;
14    }
15    return low;
16 }
17 int main()
18 {
19     int n,m;
20     scanf("%d",&n);
21     int nums[n];
22     for(int i=0;i<n;i++)
23         scanf("%d",&nums[i]);
24     scanf("%d",&m);
25     int maxes[m];
26     for(int i=0;i<m;i++)
27         scanf("%d",&maxes[i]);
28     qsort(nums,n,sizeof(int),compare);
29     for(int i=0;i<m;i++)
30     {
31         int count=upper_bound(nums,n,maxes[i]);
32         printf("%d\n",count);
33     }
34     return 0;
35 }

```

20 |

[ ]

	Input	Expected	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! ✓

//