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 != 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 %)

```
#include <stdio.h>

int main() {
    int T;
```

```c
    scanf("%d", &T);  // Number of test cases

    while (T--) {
        int N;
        scanf("%d", &N);  // Number of elements in the array

        int A[N];
        for (int i = 0; i < N; i++)
            scanf("%d", &A[i]);  // Input sorted array

        int k;
        scanf("%d", &k);  // Input value of k

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

        // Use two-pointer technique since array is sorted
        while (i < N && j < N) {
            int diff = A[j] - A[i];

            if (i != j && diff == k) {
                found = 1;
                break;
            }
            else if (diff < k)
                j++;
            else
                i++;
        }

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

    return 0;
}
```

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

	Input	Expected	Got	
✓	4 3 1 3 5 4 3 1 3 5 99 6 2 7 10 15 57 246 47 6 2 7 10 15 57 246 999	1 0 1 0	1 0 1 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 Ni (where 1 ≤ x ≤ N ≤ 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 ith test case as an integer, Ni (the number of days).

Constraints

1 ≤ T ≤ 2 × 10^5
1 ≤ N ≤ 2 × 10^6
1 ≤ x ≤ N ≤ 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

```
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 %)

```c
#include <stdio.h>

void calculate(int arr[], int T) {
    for (int i = 0; i < T; i++) {
        long long N = arr[i];
        long long k = (N + 1) / 2;  // number of odd days up to N
        printf("%lld\n", k * k);    // sum of first k odd numbers = k^2
    }
}

int main() {
    int T;
    scanf("%d", &T);
    int arr[T];
    for (int i = 0; i < T; i++) {
        scanf("%d", &arr[i]);
    }
    calculate(arr, T);
    return 0;
}
```

	Input	Expected	Got	
✓	3 1 2 3	1 1 4	1 1 4	✓
✓	10 71 100 86 54 40 9 77 9 13 98	1296 2500 1849 729 400 25 1521 25 49 2401	1296 2500 1849 729 400 25 1521 25 49 2401	✓
✓	100 540 925 332 760 482 265 610 255 347 869 139 653 514 56 992 105 292 911 875 648 122 940 774 627 484 979 87 136 730	72900 214369 27556 144400 58081 17689 93025 16384 30276 189225 4900 106929 66049 784 246016 2809 21316 207936 191844 104976 3721 220900 149769 98596 58564 240100 1936 4624 133225 99225	72900 214369 27556 144400 58081 17689 93025 16384 30276 189225 4900 106929 66049 784 246016 2809 21316 207936 191844 104976 3721 220900 149769 98596 58564 240100 1936 4624 133225 99225	✓

	Input	Expected	Got	
	629	160000	160000	
	799	35721	35721	
	378	112225	112225	
	670	3844	3844	
	123	95481	95481	
	618	153664	153664	
	784	106929	106929	
	653	42025	42025	
	409	25921	25921	
	322	163216	163216	
	808	210681	210681	
	917	125316	125316	
	708	108900	108900	
	660	75625	75625	
	550	194481	194481	
	881	19881	19881	
	281	1444	1444	
	76	123201	123201	
	701	57600	57600	
	479	8100	8100	
	180	81796	81796	
	571	230400	230400	
	960	151321	151321	
	778	164025	164025	
	810	961	961	
	62	203401	203401	
	901	40804	40804	
	403	196	196	
	27	220900	220900	
	939	214369	214369	
	926	97969	97969	
	626	153664	153664	
	783	4489	4489	
	133	182329	182329	
	854	2116	2116	
	91	2500	2500	
	100	73441	73441	
	542	196249	196249	
	885	100	100	
	20	16641	16641	
	257	86436	86436	
	588	135424	135424	
	736	47089	47089	
	434	1444	1444	
	75	114244	114244	
	676	1600	1600	
	80	171396	171396	
	828	14641	14641	

	Input	Expected	Got	
	241 243 357 831 549 172 926 600 730 51 152 184 189 856 377 881 449 55 544 607 971 733 802	14884 32041 173056 75625 7396 214369 90000 133225 676 5776 8464 9025 183184 35721 194481 50625 784 73984 92416 236196 134689 160801	14884 32041 173056 75625 7396 214369 90000 133225 676 5776 8464 9025 183184 35721 194481 50625 784 73984 92416 236196 134689 160801	

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 nums[j] ≤ maxes[i] where 0 ≤ j < n and 0 ≤ 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 ≤ 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 ≤ 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, 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 ≤ 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
5
4
8
4
3
1
7
```

8

Sample Output 1

1
0
3
4

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 ≤ 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].
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 ≤ maxes[3].

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

// Function for comparison (used in qsort)
int compare(const void *a, const void *b) {
    return (*(int *)a - *(int *)b);
}

// Function to find count of elements <= key using binary search
int upper_bound(int arr[], int n, int key) {
    int low = 0, high = n;
    while (low < high) {
        int mid = (low + high) / 2;
        if (arr[mid] <= key)
            low = mid + 1;
        else
            high = mid;
    }
    return low;
}

int main() {
    int n, m;
    scanf("%d", &n);
    int nums[n];
    for (int i = 0; i < n; i++)
    for (int i = 0; i < n; i++)
        scanf("%d", &nums[i]);

    scanf("%d", &m);
    int maxes[m];
    for (int i = 0; i < m; i++)
        scanf("%d", &maxes[i]);

    // Sort nums for efficient binary search
    qsort(nums, n, sizeof(int), compare);

    // For each maxes[i], count numbers <= maxes[i] using binary search
    for (int i = 0; i < m; i++) {
        int count = upper_bound(nums, n, maxes[i]);
        printf("%d\n", count);
    }

    return 0;
}
```

	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	✓
✓	7 23 4 2	3 6 7 0	3 6 7 0	✓

	Input	Expected	Got	
	12 18 20 16 4 13 22 29 1			

Passed all tests! ✓