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.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        int a[n];
        for(int i=0;i<n;i++)
        {
            scanf("%d",&a[i]);
        }
        int k;
        scanf("%d",&k);
        int c=0;
        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(a[i]-a[j]==k || a[j]-a[i]==k)
                {
                   c=1;
                   break;
                }
            }
            if(c)
            break;
        }
        printf("%d\n",c);
    }
}
```

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

Passed all tests! ✓

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 $1 \leq x \leq N \leq Y$) 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 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).

Answer: (penalty regime: 0 %)

```c
#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;
        }printf("%d\n",c);
    }
}
```

	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	✓

Passed all tests! ✓

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int s1,s2,ans;
    scanf("%d",&s1);
    int ta[s1];
    for(int i=0;i<s1;i++)
    scanf("%d",&ta[i]);
    scanf("%d",&s2);
    int tb[s2];
    for(int i=0;i<s2;i++)
    scanf("%d",&tb[i]);
    for(int j=0;j<s2;j++)
    {
        ans=0;
        for(int i=0;i<s1;i++)
        {
            if(tb[j]>=ta[i])
            ans++;
        }
        printf("%d\n",ans);
    }
}
```

	Input	Expected	Got	
✓	4 1 4 2 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! ✓