Problem A. Eating Queries

Time limit 3500 ms
Mem limit 262144 kB

Timur has n candies. The i-th candy has a quantity of sugar equal to a_i. So, by eating the i-th candy, Timur consumes a quantity of sugar equal to a_i.

Timur will ask you q queries regarding his candies. For the j-th query you have to answer what is the **minimum** number of candies he needs to eat in order to reach a quantity of sugar **greater than or equal to** x_j or print -1 if it's not possible to obtain such a quantity. In other words, you should print the minimum possible k such that after eating k candies, Timur consumes a quantity of sugar of at least x_j or say that no possible k exists.

Note that he can't eat the same candy twice and queries are independent of each other (Timur can use the same candy in different queries).

Input

The first line of input contains a single integer t ($1 \le t \le 1000$) — the number of test cases. The description of test cases follows.

The first line contains 2 integers n and q ($1 \le n, q \le 1.5 \cdot 10^5$) — the number of candies Timur has and the number of queries you have to print an answer for respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^4$) — the quantity of sugar in each of the candies respectively.

Then q lines follow.

Each of the next q lines contains a single integer x_j ($1 \le x_j \le 2 \cdot 10^9$) – the quantity Timur wants to reach for the given query.

It is guaranteed that the sum of n and the sum of q over all test cases do not exceed $1.5 \cdot 10^5$.

Output

For each test case output q lines. For the j-th line output the number of candies Timur needs to eat in order to reach a quantity of sugar greater than or equal to x_j or print -1 if

it's not possible to obtain such a quantity.

Examples

Input	Output
3 8 7 4 3 3 1 1 4 5 9 1 10 50 14 15 22 30 4 1 1 2 3 4 3 1 2 5 4 6	1 2 -1 2 3 4 8 1 1 -1

Note

For the first test case:

For the first query, Timur can eat any candy, and he will reach the desired quantity.

For the second query, Timur can reach a quantity of at least 10 by eating the 7-th and the 8-th candies, thus consuming a quantity of sugar equal to 14.

For the third query, there is no possible answer.

For the fourth query, Timur can reach a quantity of at least 14 by eating the 7-th and the 8-th candies, thus consuming a quantity of sugar equal to 14.

For the second test case:

For the only query of the second test case, we can choose the third candy from which Timur receives exactly 3 sugar. It's also possible to obtain the same answer by choosing the fourth candy.

Problem B. Interesting drink

Time limit 2000 ms
Mem limit 262144 kB

Vasiliy likes to rest after a hard work, so you may often meet him in some bar nearby. As all programmers do, he loves the famous drink "Beecola", which can be bought in n different shops in the city. It's known that the price of one bottle in the shop i is equal to x_i coins.

Vasiliy plans to buy his favorite drink for q consecutive days. He knows, that on the i-th day he will be able to spent m_i coins. Now, for each of the days he want to know in how many different shops he can buy a bottle of "Beecola".

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the number of shops in the city that sell Vasiliy's favourite drink.

The second line contains n integers x_i ($1 \le x_i \le 100\,000$) — prices of the bottles of the drink in the i-th shop.

The third line contains a single integer q ($1 \le q \le 100\,000$) — the number of days Vasiliy plans to buy the drink.

Then follow q lines each containing one integer m_i ($1 \le m_i \le 10^9$) — the number of coins Vasiliy can spent on the i-th day.

Output

Print q integers. The i-th of them should be equal to the number of shops where Vasiliy will be able to buy a bottle of the drink on the i-th day.

Examples

Input	Output
5 3 10 8 6 11 4 1 10 3 11	0 4 1 5

Note

On the first day, Vasiliy won't be able to buy a drink in any of the shops.

On the second day, Vasiliy can buy a drink in the shops 1, 2, 3 and 4.

On the third day, Vasiliy can buy a drink only in the shop number 1.

Finally, on the last day Vasiliy can buy a drink in any shop.

Problem C. Worms

Time limit	1000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

It is lunch time for Mole. His friend, Marmot, prepared him a nice game for lunch.

Marmot brought Mole n ordered piles of worms such that i-th pile contains a_i worms. He labeled all these worms with consecutive integers: worms in first pile are labeled with numbers 1 to a_1, worms in second pile are labeled with numbers $a_1 + 1$ to $a_1 + a_2$ and so on. See the example for a better understanding.

Mole can't eat all the worms (Marmot brought a lot) and, as we all know, Mole is blind, so Marmot tells him the labels of the best juicy worms. Marmot will only give Mole a worm if Mole says correctly in which pile this worm is contained.

Poor Mole asks for your help. For all juicy worms said by Marmot, tell Mole the correct answers.

Input

The first line contains a single integer n ($1 \le n \le 10^5$), the number of piles.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^3$, $a_1 + a_2 + \ldots + a_n \le 10^6$), where a_i is the number of worms in the i-th pile.

The third line contains single integer m ($1 \le m \le 10^5$), the number of juicy worms said by Marmot.

The fourth line contains m integers $q_1, q_2, \ldots, q_m$ ($1 \le q_i \le a_1 + a_2 + \ldots + a_n$), the labels of the juicy worms.

Output

Print m lines to the standard output. The i-th line should contain an integer, representing

the number of the pile where the worm labeled with the number q_i is.

Examples

Input	Output
5 2 7 3 4 9 3 1 25 11	1 5 3

Note

For the sample input:

- The worms with labels from $[1, 2]$ are in the first pile.
- The worms with labels from $[3, 9]$ are in the second pile.
- The worms with labels from $[10, 12]$ are in the third pile.
- The worms with labels from $[13, 16]$ are in the fourth pile.
- The worms with labels from $[17, 25]$ are in the fifth pile.

Raju and Meena love to play with Marbles. They have got a lot of marbles with numbers written on them. At the beginning, Raju would place the marbles one after another in ascending order of the numbers written on them. Then Meena would ask Raju to find the first marble with a certain number. She would count 1...2...3. Raju gets one point for correct answer, and Meena gets the point if Raju fails. After some fixed number of trials the game ends and the player with maximum points wins. Today it's your chance to play as Raju. Being the smart kid, you'd be taking the favor of a computer. But don't underestimate Meena, she had written a program to keep track how much time you're taking to give all the answers. So now you have to write a program, which will help you in your role as Raju.

Input

There can be multiple test cases. Total no of test cases is less than 65. Each test case consists begins with 2 integers: N the number of marbles and Q the number of queries Mina would make. The next N lines would contain the numbers written on the N marbles. These marble numbers will not come in any particular order. Following Q lines will have Q queries. Be assured, none of the input numbers are greater than 10000 and none of them are negative.

Input is terminated by a test case where $N = 0$ and $Q = 0$.

Output

For each test case output the serial number of the case.

For each of the queries, print one line of output. The format of this line will depend upon whether or not the query number is written upon any of the marbles. The two different formats are described below:

- 'x `found at` y', if the first marble with number x was found at position y. Positions are numbered $1, 2, \ldots, N$.
- 'x `not found`', if the marble with number x is not present.

Look at the output for sample input for details.

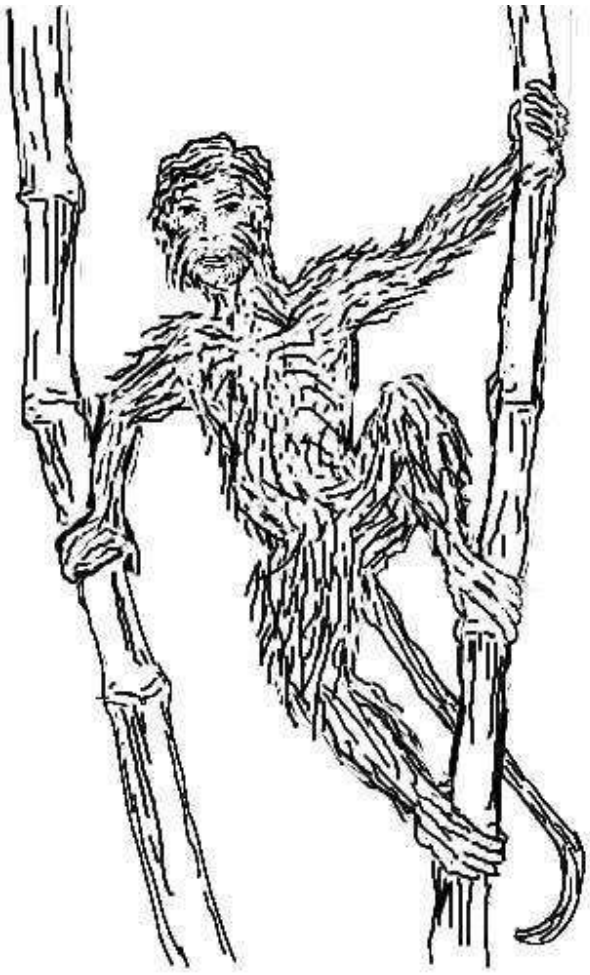
Sample Input

```
4 1
2
3
5
1
5
5 2
1
3
3
3
1
2
3
0 0
```

Sample Output

```
CASE# 1:
5 found at 4
CASE# 2:
2 not found
3 found at 3
```

Once upon a time, there lived a chimpanzee called *Luchu Bandor (aka Playboy Chimp)*. Luchu was unhappily married to Bunty Mona, a short but cute little lady chimp. Luchu was tall and handsome — he was feeling uncomfortable taking Bunty to public places along with him. People would stare at them all the while. At one point, Luchu could not stand it anymore and he decided to do some justice to his name. He started looking for *a new hope* in the Lady Chimps' High School. Every day Luchu would climb up a bamboo tree and wait for the morning drill to start. From there he could see each and every lady chimp doing their routine drill. Now, Luchu was looking for the tallest lady chimp that would be shorter than he; he would also like to consider someone a little taller than he. But someone of his same height will never be on his list. Every morning Luchu picks up a line of lady chimps and finds the best two according to his set criterion. His job has been made easy by the fact that the lady chimps in each line are ordered by their height, the shortest one is in the front and the tallest one is at the back. Your task is to help Luchu on one particular day to find two lady chimps: the tallest one shorter than he and the shortest one taller than he.



Input

There will be only one set of input for this problem. The first line of input gives you a number N ($1 \leq N \leq 50000$), the number of lady chimps on the line. In the next line you would have N integers (in the range 1 to $2^{31} - 1$ giving the heights of the N chimps. There would be a single space after every number. You can assume that the chimps are ordered in non-decreasing order of their heights. In the next line you would have an integer Q ($1 \leq Q \leq 25000$) giving the number of queries. Then in the next line Q queries will follow. Then you would have Q numbers giving the height of Luchu! Dont worry, Luchu is from the land where people can have 3 birthdates; Q heights for a chimpanzee will make no difference here. The Q numbers are listed on a line and their range from 1 to $2^{31} - 1$, and as before you would find a single space after every query number. The query numbers are not supposed to come in any particular order.

Output

For each query height, print two numbers in one line. The first one would be the height of the tallest lady chimp that is shorter than Luchu, and the next number would be the height of the shortest lady chimp that is taller than he. These two numbers are to be separated by a single space. Whenever it is impossible to find any of these two heights, replace that height with an uppercase '`X`'.

Sample Input

```
4
1 4 5 7
4
4 6 8 10
```

Sample Output

```
1 5
5 7
7 X
7 X
```

Binary Search-1

Problems:

1.

A. Binary Search

time limit per test: 2 seconds
memory limit per test: 512 megabytes

Implement a binary search algorithm.

Input

The first line of the input contains integers n and k ($1 \le n, k \le 10^5$), the length of the array and the number of queries. The second line contains n elements of the array, sorted in non-decreasing order. The third line contains k queries. All array elements and queries are integers, each of which does not exceed 10^9 in absolute value.

Output

For each of the k queries print `YES` in a separate line if this number occurs in the array, and `NO` otherwise.

Example

input Copy

```
10 10
1 61 126 217 2876 6127 39162 98126 712687 1000000000
100 6127 1 61 200 -10000 1 217 10000 1000000000
```

output Copy

```
NO
YES
YES
YES
NO
NO
YES
YES
NO
YES
```

2.

B. Closest to the Left

time limit per test: 2 seconds
memory limit per test: 512 megabytes

Given an array of n numbers, sorted in non-decreasing order, and k queries. For each query, print the maximum index of an array element not greater than the given one.

Input

The first line of the input contains integers n and k ($0 < n, k \leq 10^5$), the length of the array and the number of queries. The second line contains n elements of the array, sorted in non-decreasing order. The third line contains k queries. All array elements and queries are integers, each of which does not exceed $2 \cdot 10^9$ in absolute value.

Output

For each of the k queries, print the maximum index of an array element not greater than the given one. If there are none, print `0`.

Example

input Copy

```
5 5
3 3 5 8 9
2 4 8 1 10
```

output Copy

```
0
2
4
0
5
```

3.

C. Closest to the Right

time limit per test: 2 seconds
memory limit per test: 512 megabytes

Given an array of n numbers, sorted in non-decreasing order, and k queries. For each query, print the minimum index of an array element not less than the given one.

Input

The first line of the input contains integers n and k ($0 < n, k \leq 10^5$), the length of the array and the number of queries. The second line contains n elements of the array, sorted in non-decreasing order. The third line contains k queries. All array elements and queries are integers, each of which does not exceed $2 \cdot 10^9$ in absolute value.

Output

For each of the k queries, print the minimum index of an array element not less than the given one. If there are none, print $n + 1$.

Example

input Copy

```
5 5
3 3 5 8 9
2 4 8 1 10
```

output Copy

```
1
3
4
1
6
```

4.

D. Fast search

time limit per test: 2 seconds
memory limit per test: 512 megabytes

You are given an array a of n integers $a_1, a_2, \ldots, a_n$.

Your task is to response to the queries like "How many numbers' values are between l and r?".

Input

The first line of the input contains n — the length of the array ($1 \leq n \leq 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$).

The third line contains integer k — the number of queries ($1 \leq k \leq 10^5$).

The following k lines contain a pair of integers l r — query, described above ($-10^9 \leq l \leq r \leq 10^9$).

Output

The output must consist of k integers — responses for the queries.

Example

input Copy

```
5
10 1 10 3 4
4
1 10
2 9
3 4
2 2
```

output Copy

```
5 2 2 0
```