# Binary Search Near Repeat

*Time limit: 1 sec*

Given a sorted array of **N** elements of *possibly non-distinct* positive integers, our task is to process **M** queries. Each query is a number **Xi** and we have to find the *largest index* of the member of A not exceeding **Xi**.

The index of the array starts at 0.

## Input

- The first line of input contains two integers **N** and **M** $(1 \leq N, M \leq 100\,000)$ that give the number of element in the array and the number of queries.
- The second line contains **N** positive integer not more than 1,000,000 in non-decreasing order representing the elements in the array.
- The third line contains **M** integers representing the queries.

## Output

There must be exactly **M** lines. Each line must contain the answer of each query where it is either the largest element of the array not exceeding **Xi** or a value of -1 indicating that no such element can be found.

## Remark

For this problem, you shall not use the standard library of the languages that provides binary search, hashing or sorting, either directly or indirectly.

## Example

| Input | Output |
|---|---|
| 10  8<br>10  13  14  14  14  15  16  16  18  200<br>9  10  11  14  0  200  20  16 | -1<br>0<br>0<br>4<br>-1<br>9<br>8<br>7 |