

# Bob and his towers

Bytie has got a set of wooden blocks for his birthday. The blocks are indistinguishable from one another, as they are all cubes of the same size. Bytie forms piles by putting one block atop another. Soon he had a whole rank of such piles, one next to another in a straight line. Of course, the piles can have different heights.

Bytie's father, Byteasar, gave his son a puzzle. He gave him a number  $k$  and asked to rearrange the blocks in such a way that the number of successive piles of height at least  $k$  is maximised. However, Bytie is only ever allowed to pick the top block from a pile strictly higher than  $k$  and place it atop one of the piles next to it. Further, Bytie is not allowed to form new piles, he can only move blocks between those already existing.

## Input

In the first line of the standard input there are two integers separated by a single space:  $n$  ( $1 \leq n \leq 1\,000\,000$ ), denoting the number of piles, and  $m$  ( $1 \leq m \leq 50$ ), denoting the number of Byteasar's requests. The piles are numbered from 1 to  $n$ . In the second line there are  $n$  integers  $x_1, x_2, \dots, x_n$  separated by single spaces ( $1 \leq x_i \leq 1\,000\,000\,000$ ). The number  $x_i$  denotes the height of the  $i$ -th pile. The third line holds  $m$  integers  $k_1, k_2, \dots, k_m$  separated by single spaces ( $1 \leq k_i \leq 1\,000\,000\,000$ ). These are the subsequent values of the parameter  $k$  for which the puzzle is to be solved. That is, the largest number of successive piles of height at least  $k$  that can be obtained by allowed moves is to be determined for each given value of the parameter  $k$ .

## Output

Your program should print out  $m$  integers, separated by single spaces, to the standard output - the  $i$ -th of which should be the answer to the puzzle for the given initial piles set-up and the parameter  $k_i$ .

## Example

For the input data:

```
5 6
1 2 1 1 5
1 2 3 4 5 6
```

the correct result is:

```
5 5 2 1 1 0
```