501 Black Box

Our Black Box represents a primitive database. It can save an integer array and has a special i variable. At the initial moment Black Box is empty and i equals 0. This Black Box processes a sequence of commands (transactions). There are two types of transactions:

- ADD(x): put element x into Black Box;
- GET: increase i by 1 and give an i-minimum out of all integers containing in the Black Box.

Keep in mind that i-minimum is a number located at i-th place after Black Box elements sorting by non-descending.

Example

Let us examine a possible sequence of 11 transactions:

N	Transaction	i	Black Box contents after transaction (elements are arranged by non-descending)	Answer
1	ADD(3)	0	3	
2	GET	1	**3**	3
3	ADD(1)	1	1, 3	
4	GET	2	1, **3**	3
5	ADD(-4)	2	-4, 1, 3	
6	ADD(2)	2	-4, 1, 2, 3	
7	ADD(8)	2	-4, 1, 2, 3, 8	
8	ADD(-1000)	2	-1000, -4, 1, 2, 3, 8	
9	GET	3	-1000, -4, **1**, 2, 3, 8	1
10	GET	4	-1000, -4, 1, **2**, 3, 8	2
11	ADD(2)	4	-1000, -4, 1, 2, 2, 3, 8	

It is required to work out an efficient algorithm which treats a given sequence of transactions. The maximum number of ADD and GET transactions: 30000 of each type.

Let us describe the sequence of transactions by two integer arrays:

1. $A(1), A(2), \ldots, A(M)$: a sequence of elements which are being included into Black Box. A values are integers not exceeding 2 000 000 000 by their absolute value, $M \leq 30000$. For the Example we have $A = (3, 1, -4, 2, 8, -1000, 2)$.
2. $u(1), u(2), \ldots, u(N)$: a sequence setting a number of elements which are being included into Black Box at the moment of first, second, ... and N-transaction GET. For the Example we have $u = (1, 2, 6, 6)$.

The Black Box algorithm supposes that natural number sequence $u(1), u(2), \ldots, u(N)$ is sorted in non-descending order, $N \leq M$ and for each p $(1 \leq p \leq N)$ an inequality $p \leq u(p) \leq M$ is valid. It follows from the fact that for the p-element of our u sequence we perform a GET transaction giving p-minimum number from our $A(1), A(2), \ldots, A(u(p))$ sequence.

Input

Input data file contains (in given order): $M, N, A(1), A(2), \ldots, A(M), u(1), u(2), \ldots, u(N)$. All numbers are divided by spaces and (or) carriage return characters.

Output

Write to the output data file Black Box answers sequence for a given sequence of transactions. The numbers can be separated with spaces and end-of-line characters.

Sample Input

```
7 4
3 1 -4 2 8 -1000 2
1 2 6 6
```

Sample Output

```
3
3
1
2
```