





PEKING UNIVERSITY

JUDGE ONLINE FOR ACM/ICPC



Online Judge	Problem Set	Authors	Online Contests	User
Web Board Home Page F.A.Qs Statistical Charts	Problems Submit Problem Online Status Prob.ID: <input style="width: 50px;" type="text"/> <input type="button" value="Go"/>	Register Update your info Authors ranklist <input style="width: 50px;" type="text"/> <input type="button" value="Search"/>	Current Contest Past Contests Scheduled Contests Award Contest	User ID: <input style="width: 50px;" type="text"/> Password: <input style="width: 50px;" type="password"/> <input type="button" value="login"/> Register

Black Box

Time Limit: 1000MS Memory Limit: 10000K

Total Submissions: 13567 Accepted: 5532

Language: Default

Description

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.

Let us examine a possible sequence of 11 transactions:

Example 1

N Transaction *i* Black Box contents after transaction Answer

(elements are arranged by non-descending)

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. $A(1), A(2), \dots, A(M)$: 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. $u(1), u(2), \dots, u(N)$: 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 $u(1), u(2), \dots, u(N)$ 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 $A(1), A(2), \dots, A(u(p))$ sequence.

Input

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

Output

Write to the output Black Box answers sequence for a given sequence of transactions, one number each line.

Sample Input

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

Sample Output

```
3
3
1
2
```

Source

Northeastern Europe 1996

[\[Submit\]](#) [\[Go Back\]](#) [\[Status\]](#) [\[Discuss\]](#)



[Home Page](#)



[Go Back](#)



[To top](#)

All Rights Reserved 2003-2013 Ying Fuchen,Xu Pengcheng,Xie Di
Any problem, Please [Contact Administrator](#)