Sherlock and Watson

John Watson performs an operation called *Right Circular Rotation* on an integer array $[a_0, a_1, \ldots a_{N-1}]$. *Right Circular Rotation* transforms the array from $[a_0, a_1, \ldots a_{N-1}]$ to $[a_{N-1}, a_0, \ldots, a_{N-2}]$.

He performs the operation K times and tests Sherlock's ability to identify the element at a particular position in the array. He asks Q queries. Each query consists of one integer, idx, for which you have to print the element at index idx in the rotated array, i.e. a_{idx}.

Input Format
The first line consists of three integers, N, K, and Q,, separated by a single space.
The next line contains N space-separated integers which indicate the elements of the array A.
Each of the next Q lines contains one integer per line denoting idx.

Output Format
For each query, print the value at index idx in the updated array separated by newline.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq a_i \leq 10^5$
- $1 \leq K \leq 10^5$
- $1 \leq Q \leq 500$
- $0 \leq idx \leq N-1$

Sample input

```
3 2 3
1 2 3
0
1
2
```

Sample output

```
2
3
1
```

Explanation
After one rotation array becomes, *[3, 1, 2]*.
After another rotation array becomes *[2, 3, 1]*.
Final array now is *[2, 3, 1]*.
0^{th} element of array is 2.
1^{st} element of array is 3.
2^{nd} element of array is 1.