# Sherlock and Watson

John Watson performs an operation called *Right Circular Rotation* on an integer array $[a$_{$0$}$, a$_{$1$}$, ... a$_{$N-1$}$]. *Right Circular Rotation* transforms the array from $[a$_{$0$}$, a$_{$1$}$, ... a$_{$N-1$}$] to $[a$_{$N-1$}$, a$_{$0$}$, ..., a$_{$N-2$}$].

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**

- $1 \le N \le 10$^{$5$}$
- $1 \le a$_{$i$}$ $\le 10$^{$5$}$
- $1 \le K \le 10$^{$5$}$
- $1 \le Q \le 500$
- $0 \le idx \le N-1$

**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.