# Circular Array Rotation

John Watson knows of an operation called a *right circular rotation* on an array of integers. One rotation operation moves the last array element to the first position and shifts all remaining elements right one. To test Sherlock's abilities, Watson provides Sherlock with an array of integers. Sherlock is to perform the rotation operation a number of times then determine the value of the element at a given position.

For each array, perform a number of right circular rotations and return the value of the element at a given index.

**Input Format**

The first line contains $3$ space-separated integers, $n$, $k$, and $q$.
The second line contains $n$ space-separated integers, where each integer $i$ describes array element $a_i$ (where $0 \leq i < n$).
Each of the $q$ subsequent lines contains a single integer denoting $m$.

**Constraints**

- $1 \leq n \leq 10^5$
- $1 \leq a_i \leq 10^5$
- $1 \leq k \leq 10^5$
- $1 \leq q \leq 500$
- $0 \leq m < n$

**Output Format**

For each query, print the value of the element at index $m$ of the rotated array on a new line.

**Sample Input 0**

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

**Sample Output 0**

```
2
3
1
```

**Explanation 0**

After the first rotation, the array becomes $[3, 1, 2]$.
After the second (and final) rotation, the array becomes $[2, 3, 1]$.

Let's refer to the array's final state as array $b = [2, 3, 1]$. For each query, we just have to print the value of $b_m$ on a new line:

1. $m = 0$, $b_0 = 2$.
2. $m = 1$, $b_1 = 3$.

3. $m = 2$, $b_2 = 1$.