# Circular Array Rotation

John Watson performs an operation called a *right circular rotation* on an array of integers, $[a_0, a_1, \ldots a_{n-1}]$. After performing one *right circular rotation* operation, the array is transformed from $[a_0, a_1, \ldots a_{n-1}]$ to $[a_{n-1}, a_0, \ldots, a_{n-2}]$.

Watson performs this operation $k$ times. To test Sherlock's ability to identify the current element at a particular position in the rotated array, Watson asks $q$ queries, where each query consists of a single integer, $m$, for which you must print the element at index $m$ in the rotated array (i.e., the value of $a_m$).

### Input Format

The first line contains $3$ space-separated integers, $n$, $k$, and $q$, respectively.
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

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

### Output Format

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

### Sample Input

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

### Sample Output

```
2
3
1
```

### Explanation

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 $b$. For each query, we just have to print the value of $b_m$ on a new line:

1. $m = 0$, so we print $2$ on a new line.
2. $m = 1$, so we print $3$ on a new line.
3. $m = 2$, so we print $1$ on a new line.