**Problem: Circular Array Rotation**

John Watson performs an operation called a *right circular rotation* on an array of integers, . After performing one *right circular rotation* operation, the array is transformed from  to .

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

**Input Format**

The first line contains  space-separated integers, , , and , respectively.   
The second line contains  space-separated integers, where each integer  describes array element  (where ).   
Each of the  subsequent lines contains a single integer denoting .

**Constraints**

**Output Format**

For each query, print the value of the element at index  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 .   
After the second (and final) rotation, the array becomes .

Let's refer to the array's final state as array . For each query, we just have to print the value of  on a new line:

1. , so we print  on a new line.
2. , so we print  on a new line.
3. , so we print  on a new line.

**Solution**

int main()

{

long size, queries, rotations, index;

cin>>size >>rotations >>queries;

long array[size];

/\*Feeding the data\*/

for(long i=0; i<size; i++)

{

cin>>array[i];

}

/\*Handling the queries\*/

for(int i=0; i<queries; i++)

{

cin>>index;

cout<<array[ (index + rotations\*(size-1) )%size]<<endl;

}

return 0;

}

* Anshul Aggarwal