Circular Array Rotation

49 more points to get your gold badge!

Rank: 275224 | Points: 801/850

Your Circular Array Rotation submission got 20.00 points.

You are now 49 points away from the gold level for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

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 values of the elements at the given indices.

Example

$a = [3, 4, 5]$

$k = 2$

$queries = [1, 2]$

Here k is the number of rotations on a, and $queries$ holds the list of indices to report. First we perform the two rotations: $[3, 4, 5] \rightarrow [5, 3, 4] \rightarrow [4, 5, 3]$

Now return the values from the zero-based indices 1 and 2 as indicated in the $queries$ array.

$a[1] = 5$

$a[2] = 3$

Function Description

Complete the circularArrayRotation function in the editor below.

circularArrayRotation has the following parameter(s):

- int a[n]: the array to rotate
- int k: the rotation count
- int queries[l]: the indices to report

Returns

- int[q]: the values in the rotated a as requested in m

Input Format

The first line contains 3 space-separated integers, n, k, and q, the number of elements in the integer array, the rotation count and the number of queries.

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, $queries[i]$, an index of an element in a to return.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq a[i] \leq 10^5$
- $1 \leq k \leq 10^5$
- $1 \leq q \leq 500$
- $0 \leq queries[i] < n$

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 is $[3, 1, 2]$.

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

We will call this final state array $b = [2, 3, 1]$. For each query, we just have to get the value of $b[queries[i]]$.

1. $queries[0] = 0$, $b[0] = 2$.
2. $queries[1] = 1$, $b[1] = 3$.
3. $queries[2] = 2$, $b[2] = 1$.

Change Theme Language C++14

```cpp
/* The function is expected to return an INTEGER_ARRAY.
 * The function accepts following parameters:
 *  1. INTEGER_ARRAY a
 *  2. INTEGER k
 *  3. INTEGER_ARRAY queries
 */

vector<int> circularArrayRotation(vector<int> a, int k, vector<int> queries) {
    vector<int> newArray(a.size()), result;
    for(int i = 0; i < a.size(); i++){
        newArray[(i + k) % a.size()] = a[i];
    }
    for(int i = 0; i < queries.size(); i++) result.push_back(newArray[queries[i]]);
    return result;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string first_multiple_input_temp;
    getline(cin, first_multiple_input_temp);

    vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));

    int n = stoi(first_multiple_input[0]);

    int k = stoi(first_multiple_input[1]);

    int q = stoi(first_multiple_input[2]);
```

Line: 122 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!

You are now 49 points away from the gold level for your problem solving badge.

87% 801/850

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5
- Test case 6

Compiler Message

```
Success
```

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock