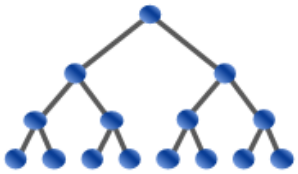
USA Computing Olympiad



USACO 2023 US Open Contest, Bronze
Problem 3. Rotate and Shift

Return to Problem List

Contest has ended.

Submitted; Results below show the outcome for each judge test case

1	2	3	4	5	6	7	8	9	10	11	12	13
* 36.3mb 317ms	* 37.4mb 459ms	* 39.1mb 551ms	* 37.4mb 469ms	* 37.3mb 462ms	* 38.8mb 567ms	* 37.6mb 511ms	* 58.2mb 2507ms	* 56.2mb 2672ms	* 60.0mb 2839ms	* 56.3mb 2678ms	* 56.6mb 2593ms	* 62.2mb 3206ms

English (en)

****Note: The time limit for this problem is 4s, 2x the default.****

To celebrate the start of spring, Farmer John's N cows ($1 \le N \le 2 \cdot 10^5$) have invented an intriguing new dance, where they stand in a circle and re-order themselves in a predictable way.

Specifically, there are N positions around the circle, numbered sequentially from 0 to $N-1$, with position 0 following position $N-1$. A cow resides at each position. The cows are also numbered sequentially from 0 to $N-1$. Initially, cow i starts in position i. You are told a set of K positions $0 = A_1 < A_2 < \ldots < A_K < N$ that are "active", meaning the cows in these positions are the next to move ($1 \le K \le N$).

In each minute of the dance, two things happen. First, the cows in the active positions rotate: the cow at position A_1 moves to position A_2, the cow at position A_2 moves to position A_3, and so on, with the cow at position A_K moving to position A_1. All of these K moves happen simultaneously, so the after the rotation is complete, all of the active positions still contain exactly one cow. Next, the active positions themselves shift: A_1 becomes $A_1 + 1$, A_2 becomes $A_2 + 1$, and so on (if $A_i = N - 1$ for some active position, then A_i circles back around to 0).

Please calculate the order of the cows after T minutes of the dance ($1 \le T \le 10^9$).

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains three integers N, K, and T.

The second line contains K integers representing the initial set of active positions $A_1, A_2, \ldots A_K$. Recall that $A_1 = 0$ and that these are given in increasing order.

OUTPUT FORMAT (print output to the terminal / stdout):

Output the order of the cows after T minutes, starting with the cow in position 0, separated by spaces.

SAMPLE INPUT:

```
5 3 4
0 2 3
```

SAMPLE OUTPUT:

```
1 2 3 4 0
```

For the example above, here are the cow orders and A for the first four timesteps:

```
Initial, T = 0: order = [0 1 2 3 4], A = [0 2 3]
T = 1: order = [3 1 0 2 4]
T = 1: A = [1 3 4]
T = 2: order = [3 4 0 1 2]
T = 2: A = [2 4 0]
T = 3: order = [2 4 3 1 0]
```

```
T = 3: A = [3 0 1]
T = 4: order = [1 2 3 4 0]
```

SCORING:

- Inputs 2-7: $N \leq 1000, T \leq 10000$
- Inputs 8-13: No additional constraints.

Problem credits: Claire Zhang

Language: C

Source File: Choose File No file chosen

Submit Solution

Note: Many issues (e.g., uninitialized variables, out-of-bounds memory access) can cause a program to produce different output when run multiple times; if your program behaves in a manner inconsistent with the official contest results, you should probably look for one of these issues. Timing can also differ slightly from run to run, so it is possible for a program timing out in the official results to occasionally run just under the time limit in analysis mode, and vice versa. Note also that we have recently changed grading servers, and since our new servers run at different speeds from the servers used during older contests, timing results for older contest problems may be slightly off until we manage to re-calibrate everything properly.

Previous In-Contest Submissions:

Sun, Mar 26, 2023 14:39:53 EDT (Java)
Sun, Mar 26, 2023 14:45:00 EDT (Java)
Sun, Mar 26, 2023 14:53:10 EDT (Java)
Sun, Mar 26, 2023 14:56:31 EDT (Java)
Sun, Mar 26, 2023 15:04:48 EDT (Java)