Arrays: Left Rotation ★

Problem Submissions Leaderboard Editorial

A left rotation operation on an array shifts each of the array's elements 1 unit to the left. For example, if 2 left rotations are performed on array $[1, 2, 3, 4, 5]$, then the array would become $[3, 4, 5, 1, 2]$. Note that the lowest index item moves to the highest index in a rotation. This is called a circular array.

Given an array a of n integers and a number, d, perform d left rotations on the array. Return the updated array to be printed as a single line of space-separated integers.

Function Description

Complete the function rotLeft in the editor below.

rotLeft has the following parameter(s):

- int a[n]: the array to rotate
- int d: the number of rotations

Returns

- int a'[n]: the rotated array

Input Format

The first line contains two space-separated integers n and d, the size of a and the number of left rotations.
The second line contains n space-separated integers, each an $a[i]$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq d \leq n$
- $1 \leq a[i] \leq 10^6$

Sample Input

```
5 4
1 2 3 4 5
```

Sample Output

```
5 1 2 3 4
```

Explanation

When we perform $d = 4$ left rotations, the array undergoes the following sequence of changes:

$$[1, 2, 3, 4, 5] \rightarrow [2, 3, 4, 5, 1] \rightarrow [3, 4, 5, 1, 2] \rightarrow [4, 5, 1, 2, 3] \rightarrow [5, 1, 2, 3, 4]$$

Change Theme JavaScript (Node.js)

```
21  });
22
23  function readLine() {
24      return inputString[currentLine++];
25  }
26
27  // Complete the rotLeft function below.
```

```
28  function rotLeft(arr, rotations) {
29      const rotatedArray = arr.concat();
30      for (let i=0; i<rotations; i++) {
31          const frontItem = rotatedArray.shift();
32          rotatedArray.push(frontItem);
33      }
34      return rotatedArray;
35
36  }
37
38  function main() {
39      const ws = fs.createWriteStream(process.env.OUTPUT_PATH);
40
41      const nd = readLine().split(' ');
42
43      const n = parseInt(nd[0], 10);
44
```

Line: 34 Col: 25

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

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Sample Test case 2

Input (stdin) Download

```
5 4
1 2 3 4 5
```

Your Output (stdout)

```
5 1 2 3 4
```

Expected Output Download

```
5 1 2 3 4
```