Left Rotation

A ***left rotation*** operation on a circular array shifts each of the array's elements 1 unit to the left. The elements that fall off the left end reappear at the right end. Given an integer d, rotate the array that many steps to the left and return the result.

Example

$d = 2$

$arr = [1, 2, 3, 4, 5]$

After 2 rotations, $arr' = [3, 4, 5, 1, 2]$.

Function Description

Complete the ***rotateLeft*** function with the following parameters:

- ***int d***: the amount to rotate by
- ***int arr[n]***: the array to rotate

Returns

- ***int[n]***: the rotated array

Input Format

The first line contains two space-separated integers that denote n, the number of integers, and d, the number of left rotations to perform.
The second line contains n space-separated integers that describe $arr[]$.

Constraints

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

Sample Input

```
STDIN     Function
-----     --------
5 4       n = 5 d = 4
1 2 3 4 5 arr = [1, 2, 3, 4, 5]
```

Sample Output

```
5 1 2 3 4
```

Explanation

To 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 Language Python 3

```
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts following parameters:
#  1. INTEGER d
#  2. INTEGER_ARRAY arr
#

def rotateLeft(d, arr):
    # Write your code here
    arr.reverse()
    for i in range((n-d)//2):
        arr[i], arr[n-d-1-i] = arr[n-d-1-i], arr[i]
    for i in range(d//2):
        arr[n-d+i],arr[n-1-i] = arr[n-1-i],arr[n-d+i]
    return arr

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    first_multiple_input = input().rstrip().split()

    n = int(first_multiple_input[0])

    d = int(first_multiple_input[1])

    arr = list(map(int, input().rstrip().split()))

    result = rotateLeft(d, arr)

    fptr.write(' '.join(map(str, result)))
    fptr.write('\n')

    fptr.close()

```

EMACS Line: 33 Col: 1

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

You have earned 20.00 points!

You are now 9 points away from the 2nd star for your problem solving badge.

87% 91/100

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

Input (stdin) Download

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

Expected Output Download

```
5 1 2 3 4
```