Array rotaion

A *left rotation* operation on an array of size  shifts each of the array's elements  unit to the left. For example, if left rotations are performed on array , then the array would become .

Given an array of  integers and a number, , perform  left rotations on the array. Then print the updated array as a single line of space-separated integers.

**Input Format**

The first line contains two space-separated integers denoting the respective values of  (the number of integers) and  (the number of left rotations you must perform).   
The second line contains  space-separated integers describing the respective elements of the array's initial state.

**Constraints**

**Output Format**

Print a single line of  space-separated integers denoting the final state of the array after performing  left rotations.

**Sample Input**

5 4

1 2 3 4 5

def shift\_list(array, s):

"""Shifts the elements of a list to the left or right.

Args:

array - the list to shift

s - the amount to shift the list ('+': left-shift, '-': right-shift)

Returns:

shifted\_array - the shifted list

"""

# calculate actual shift amount (e.g., 11 --> 1 if length of the array is 5)

s %= len(array)

# uncomment this line to reverse the shift direction

# s \*= -1

# shift array with list slicing

shifted\_array = array[s:] + array[:s]

return shifted\_array

my\_array = [1, 2, 3, 4, 5]

# negative numbers

shift\_list(my\_array, -7)

>>> [4, 5, 1, 2, 3]

# no shift on numbers equal to the size of the array

shift\_list(my\_array, 5)

>>> [1, 2, 3, 4, 5]

# works on positive numbers

shift\_list(my\_array, 3)

>>> [4, 5, 1, 2, 3]