Check out the resources on the page's right side to learn more about arrays. The video tutorial is by Gayle Laakmann McDowell, author of the best-selling interview book [Cracking the Coding Interview](https://www.hackerrank.com/ctci-book?ref=body).

A *left rotation* operation on an array 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. 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. It should return the resulting array of integers.

rotLeft has the following parameter(s):

* An array of integers .
* An integer , the number of rotations.

**Input Format**

The first line contains two space-separated integers  and , the size of  and the number of left rotations you must perform.   
The second line contains  space-separated integers .

**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

**Sample Output**

5 1 2 3 4

**Explanation**

When we perform  left rotations, the array undergoes the following sequence of changes:

**Current Buffer** (saved locally, editable)

[Java 7](javascript:void(0))





1

import java.io.\*;

2

import java.math.\*;

3

import java.security.\*;

4

import java.text.\*;

5

import java.util.\*;

6

import java.util.concurrent.\*;

7

import java.util.regex.\*;

8

​

9

public class Solution {

10

​

11

   // Complete the rotLeft function below.

12

   static int[] rotLeft(int[] a, int d) {

13

       // Get Sub array based on value of 'd'

14

       int[] arrRotationElements = Arrays.copyOfRange(a, 0, d);

15

       int[] arrRemainingElements = Arrays.copyOfRange(a, d, a.length);

16

17

       int[] resultArray = new int[arrRotationElements.length + arrRemainingElements.length];

18

19

       System.arraycopy(arrRemainingElements, 0, resultArray, 0, arrRemainingElements.length);

20

       System.arraycopy(arrRotationElements, 0, resultArray, arrRemainingElements.length, arrRotationElements.length);

21

22

       for(int i = 0; i < resultArray.length; i++)

23

           System.out.print(resultArray[i] + " ");

24

25

       return resultArray;

26

​

27

  }

28

​

29

   private static final Scanner scanner = new Scanner(System.in);

30

​

31

   public static void main(String[] args) throws IOException {

32

       BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

33

​

34

       String[] nd = scanner.nextLine().split(" ");

35

​

36

       int n = Integer.parseInt(nd[0]);

37

​

38

       int d = Integer.parseInt(nd[1]);

39

​

40

       int[] a = new int[n];

41

​

42

       String[] aItems = scanner.nextLine().split(" ");

43

       scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

for (int i = 0; i < n; i++) {

int aItem = Integer.parseInt(aItems[i]);

a[i] = aItem;

}

int[] result = rotLeft(a, d);

for (int i = 0; i < result.length; i++) {

bufferedWriter.write(String.valueOf(result[i]));

if (i != result.length - 1) {

bufferedWriter.write(" ");

}

}

bufferedWriter.newLine();

bufferedWriter.close();

scanner.close();

}

}