**Monk and Rotation**

Attempted by: **2563**

/

Accuracy: **92%**

/

Maximum Score: **20**

/

14 Votes

Tag(s):

Data Structures, Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Monk loves to preform different operations on arrays, and so being the principal of Hackerearth School, he assigned a task to his new student Mishki. Mishki will be provided with an integer array AA of size NN and an integer KK , where she needs to rotate the array in the right direction by K steps and then print the resultant array. As she is new to the school, please help her to complete the task.  
  
**Input**:  
The first line will consists of one integer TT denoting the number of test cases.   
For each test case:  
1) The first line consists of two integers NN and KK, NN being the number of elements in the array and KK denotes the number of steps of rotation.  
2) The next line consists of NN space separated integers , denoting the elements of the array AA.

**Output**:  
Print the required array.

**Constraints**:  
1≤T≤201≤T≤20  
1≤N≤1051≤N≤105  
0≤K≤1060≤K≤106  
0≤A[i]≤1060≤A[i]≤106

**SAMPLE INPUT**

1

5 2

1 2 3 4 5

**SAMPLE OUTPUT**

4 5 1 2 3

**Explanation**

Here TT is 1, which means one test case.   
N=5N=5 denoting the number of elements in the array and K=2K=2, denoting the number of steps of rotations.   
The initial array is: 1,2,3,4,51,2,3,4,5   
 In first rotation, 55 will come in the first position and all other elements will move to one position ahead from their current position. Now, the resultant array will be 5,1,2,3,45,1,2,3,4   
In second rotation, 44 will come in the first position and all other elements will move to one position ahead from their current position. Now, the resultant array will be 4,5,1,2,34,5,1,2,3

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/monk-and-rotation-3/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int[] RotarDerecha(int[] arr, int k)

{

k = k % arr.Length;

int[] rotado = new int[arr.Length];

int i = 0;

for (i = k; i < rotado.Length; i++)

{

rotado[i] = arr[i - k];

}

for (int j = i - k; j < arr.Length; j++)

{

rotado[j - (i - k)] = arr[j];

}

return rotado;

}

static void Main(string[] args)

{

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

string[] input = Console.ReadLine().Split(' ');

int n = int.Parse(input[0]);

int k = int.Parse(input[1]);

int[] a = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

foreach (int elem in RotarDerecha(a, k))

{

Console.Write(elem + " ");

}

Console.WriteLine();

}

// Console.ReadLine();

}

}

}