codility

aining center
k out Codility training tasks

Congratulations

You have completed a Codility training test.

Tweet this!
I scored 100% in #java on @Codility!
https://codility.com/demo/take-sample-test/cyclic_rotation/

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: training29XQEW-AER
Time limit: 120 min.

Status: closed

Created on: 2016-06-18 16:50 UTC
Started on: 2016-06-18 16:50 UTC
Finished on: 2016-06-18 17:12 UTC

Tasks in test	Correctness	Performance	Task score
1 CyclicRotation Submitted in: Java	100%	not assessed	100%

Test score

100%

100 out of 100 points

EASY

1. CyclicRotation

Rotate an array to the right by a given number of steps.

score: 100 of 100

Task description

A zero-indexed array A consisting of N integers is given. Rotation of the array means that each element is shifted right by one index, and the last element of the array is also moved to the first place.

For example, the rotation of array A = [3, 8, 9, 7, 6] is [6, 3, 8, 9, 7]. The goal is to rotate array A K times; that is, each element of A will be shifted to the right by K indexes.

Write a function:

```
class Solution { public int[] solution(int[] A, int K); }
```

that, given a zero-indexed array A consisting of N integers and an integer K, returns the array A rotated K times.

For example, given array A = [3, 8, 9, 7, 6] and K = 3, the function should return [9, 7, 6, 3, 8].

Assume that:

- N and K are integers within the range [0..100];
- each element of array A is an integer within the range [−1,000..1,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Solution

Programming language used: Java

Total time used: 22 minutes

Effective time used: 22 minutes

Notes: *not defined yet*

Task timeline

16:50:42 17:12:27

Code: 17:12:27 UTC, java, final, score: **100**

show code in pop-up

```
1  import java.util.Collections;
2  import java.util.List;
3  import java.util.stream.IntStream;
4
5  import static java.util.stream.Collectors.toList;
6
```

```
7  public class Solution {
8      /**
9       *
10      * @param A - array elements to rotate
11      * @param K - rotation counter
12      * @return
13      */
14     public int[] solution(int[] A, int K) {
15         List<Integer> integers = IntStream.of(A).boxed(
16         Collections.rotate(integers, K);
17         return integers.stream().mapToInt(x -> x).toArr
18     }
19 }
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶ example example test		✔ OK
expand all	Correctness tests	
▶ extreme_empty empty array		✔ OK
▶ single one element, 0 <= K <= 5		✔ OK
▶ double two elements, K <= N		✔ OK
▶ small1 small functional tests, K < N		✔ OK
▶ small2 small functional tests, K >= N		✔ OK
▶ small_random_all_rotations small random sequence, all rotations, N = 15		✔ OK
▶ medium_random medium random sequence, N = 100		✔ OK
▶ maximal maximal N and K		✔ OK

Training center