Codility_

Screen Report: Anonymous

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Effective time spent	Score
CyclicRotation ⚠ C++	3 min	100%

Total score

100%

Tasks Details

Easy	1. CyclicRotation Rotate an array to the right by a given number of steps.	Task Score	Correctness	Performance
		100%	100%	Not assessed

Task description

An 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 moved to the first place. For example, the rotation of array A = [3, 8, 9, 7, 6] is [6, 3, 8, 9, 7] (elements are shifted right by one index and 6 is moved to the first place).

The goal is to rotate array A K times; that is, each element of A will be shifted to the right K times.

Write a function:

```
vector<int> solution(vector<int> &A, int K);
```

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

For example, given

```
A = [3, 8, 9, 7, 6]
K = 3
```

Solution

Programming language used:	C++
Time spent on task:	3 minutes ?
Notes:	*not defined yet*

Task timeline ?

07:31:14 07:33:39

Code: 07:33:39 UTC, cpp, final, score: **100** show code in pop-up the function should return [9, 7, 6, 3, 8]. Three rotations were made:

```
[3, 8, 9, 7, 6] -> [6, 3, 8, 9, 7]
[6, 3, 8, 9, 7] -> [7, 6, 3, 8, 9]
[7, 6, 3, 8, 9] -> [9, 7, 6, 3, 8]
```

For another example, given

```
A = [0, 0, 0]
K = 1
```

the function should return [0, 0, 0]

Given

```
A = [1, 2, 3, 4]
K = 4
```

the function should return [1, 2, 3, 4]

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.

```


// you can use includes, for example:
#include <vector>


vector<int> solution(vector<int> &A, int K) {
    // Implement your solution here
    if(A.empty())
    {
        return A;
    }

    int sizeOfA{static_cast<int>(A.size())};
    vector<int> rotatedA(A);
    int realRotations{K % sizeOfA};

    if(realRotations == 0)
    {
        return rotatedA;
    }

    for (int i = 0; i < sizeOfA; i++)
    {
        int mappedElementAfterRotation{(sizeOfA -
        rotatedA[i] = A[mappedElementAfterRotation
    }

    return rotatedA;

}

```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	**Example tests**	
▶	example first example test	✓ OK
▶	example2 second example test	✓ OK
▶	example3 third 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	✓ OK

small random sequence, all rotations, N = 15		
▶	medium_random medium random sequence, N = 100	✓ OK
▶	maximal maximal N and K	✓ OK