Codility_

## CodeCheck Report: trainingZN5SMW-94U

Test Name:

Check out Co

Summary Timeline

### Tasks summary

| Task | Time spent | Score |
|---|---|---|
| CyclicRotation<br>Go | 1 min | 75% |

### Total score


## Tasks Details

Easy

### 1. CyclicRotation

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

| Task Score | Correctness | Performance |
|---|---|---|
| 75% | 75% | 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:

```
func Solution(A []int, K int) []int
```

### Solution

| | |
|---|---|
| Programming language used: | Go |
| Total time used: | 1 minutes |
| Effective time used: | 1 minutes |
| Notes: | *not defined yet* |

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.



final, score: **75**

```go
package solution

import "math"

func rotateRight(A []int, K int) []i
    lastKElementsToPutInFront := A[l
    return append(lastKElementsToPut
}

func rotateLeft(A []int, K int) []in
    firstKElementsToPutInBack := A[:
    return append(A[K:], firstKEleme
}

func Solution(A []int, K int) []int
    if len(A) == 0 {
        return A
    }

    if K >= len(A) {
        K = len(A) % K
    }

    if K == 0 {
        return A
    }

    KLeft := len(A) - K

    minimumK := math.Min(float64(K),

    if minimumK == float64(K) {
        return rotateRight(A, K)
    }

    return rotateLeft(A, KLeft)
}
```

## Analysis summary

The following issues have been detected: wrong a

For example, for the input ( `[1, 1, 2, 3, 5]`, solution returned a wrong answer (got [1, 1, 2, 3, 5 5, 1, 1, 2]).

## Analysis

| expand all | Example tests | |
|---|---|---|
| ▶ | example<br>first example test | ✔ OK |
| ▶ | example2<br>second example test | ✔ OK |

| | | |
|---|---|---|
| ▶ | double<br>two elements, K <= N | ✔ OK |
| ▶ | small1<br>small functional tests, K < N | ✔ OK |
| | small2<br>small functional tests, K >= N | ✘ WRONG<br>got [-1, -2, -<br>expected [-<br>-.. |

1. 0.001 s **OK**
2. 0.001 s **WRONG ANSWER**, got [-1, -2, -3, -4, -5, -.. ex
   -5, -6, -1, -..
3. 0.001 s **WRONG ANSWER**, got [1, 1, 2, 3, 5] expecte

| | | |
|---|---|---|
| ▶ | small_random_all_rotations<br>small random sequence, all rotations, N = 15 | ✔ OK |
| ▶ | medium_random<br>medium random sequence, N = 100 | ✔ OK |
| | maximal<br>maximal N and K | ✘ WRONG<br>got [710, 8(<br>expected [1<br>568, .. |

1. 0.001 s **OK**
2. 0.001 s **WRONG ANSWER**, got [710, 807, 568, 560,
   [155, 710, 807, 568, ..
3. 0.001 s **OK**
4. 0.001 s **OK**

---