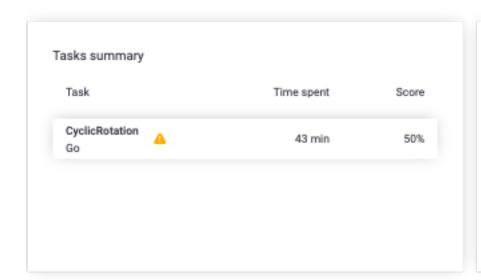
# Codility\_

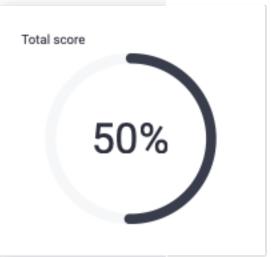
# CodeCheck Report: trainingVJ2JM6-YRG

Test Name:

Check out Co

Summary Timeline





#### **Tasks Details**

#### 1. CyclicRotation

Easy

Rotate an array to the right Task Score by a given number of 50% steps.

Correctness

Performance
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: 43 minutes

Effective time used: 43 minutes

Notes: not defined yet

1 of 5

made:

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.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
final, score: 50
```

```
1
    package solution
2
3
    import "math"
4
    func rotateRight(A []int, K int) []i
5
6
         lastKElementsToPutInFront := A[l
7
         return append(lastKElementsToPut
    }
8
9
10
    func rotateLeft(A []int, K int) []in
         firstKElementsToPutInBack := A[:
11
12
         return append(A[K:], firstKEleme
13
14
    func Solution(A []int, K int) []int
15
16
         if len(A) == K {
17
             return A
18
19
         KLeft := len(A) - K
20
21
         minimumK := math.Min(float64(K),
22
23
24
         if minimumK == float64(K) {
25
             return rotateRight(A, K)
26
         }
27
28
         return rotateLeft(A, KLeft)
    }
29
```

#### Analysis summary

The following issues have been detected: runtime

For example, for the input ([], 1) the solution t unexpectedly.

## **Analysis**

collapse all	Example tests
example first example test	<b>√</b> OK
1. 0.001 s <b>OK</b>	
example2 second example te	<b>∨ OK</b> t
1. 0.001 s <b>OK</b>	
example3 third example test	<b>∨</b> OK

```
2. 0.001 s RUNTIME ERROR, tested program termina
   stderr:
   panic: runtime error: slice bounds out
   goroutine 1 [running]:
   solution.Solution(0x573a20, 0x0, 0x0,
            single
                                ✗ RUNTIM
   one element, 0 <= K <= 5
                                  tested prog
                                  terminated
1. 0.001 s OK
2. 0.001 s OK
3. 0.001 s RUNTIME ERROR, tested program termina
         code 2
   stderr:
   panic: runtime error: slice bounds out
   goroutine 1 [running]:
   solution.Solution(0xc208032288, 0x1, (
           double
                                ✓ OK
   two elements, K <= N
1. 0.001 s OK
2. 0.001 s OK
   small1
                                OK
   small functional tests, K < N \,
1. 0.001 s OK
2. 0.001 s OK
                                × RUNTIM
   small2
   small functional tests, K >= N
                                  tested prog
                                  terminated
                                  2
1. 0.001 s OK
2. 0.001 s RUNTIME ERROR, tested program termina
          code 2
   stderr:
   panic: runtime error: slice bounds out
   goroutine 1 [running]:
   solution.Solution(0xc20800a2d0, 0x6, (
```

3. 0.001 s RUNTIME ERROR, tested program termina code 2 stderr: panic: runtime error: slice bounds out goroutine 1 [running]: solution.Solution(0xc20800a2d0, 0x5, ( small\_random\_all\_rotations small random sequence, all rotations, N = 15 1. 0.001 s **OK** 2. 0.001 s **OK** 3. 0.001 s **OK** 4. 0.001 s **OK** 5. 0.001 s **OK** 6. 0.001 s **OK** 7. 0.001 s **OK** 8. 0.001 s **OK** 9. 0.001 s **OK** 10. 0.001 s **OK** 11. 0.001 s **OK** 12. 0.001 s **OK** 13. 0.001 s **OK** 14. 0.001 s **OK** 15. 0.001 s **OK** medium\_random ✓ OK medium random sequence, N = 100 1. 0.001 s **OK** 2. 0.001 s **OK ★** RUNTIM maximal maximal N and K tested prog terminated 1. 0.001 s **OK** 2. 0.001 s RUNTIME ERROR, tested program termina code 2

3. 0.001 s **OK** 

4. 0.001 s **OK** 

5 of 5