codility

g center

lity training tasks

Congratulations

You have completed a Codility training test.

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

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingA5R3HS-528
Time limit: 120 min.

Status: closed

Created on: 2017-09-01 16:04 UTC
Started on: 2017-09-01 16:04 UTC
Finished on: 2017-09-01 16:09 UTC

Tasks in test	Correctness	Performance	Task score
1 <·> CyclicRotation Submitted in: Python	100%	not assessed	100%

Test score ?

100%

100 out of 100 points

How likely are you to recommend Codility to your friends and colleagues?

X

Not at all likely *Extremely likely*

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:

```
def solution(A, 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

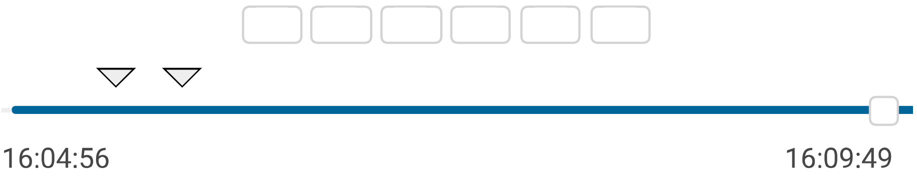
Programming language used: Python

Total time used: 5 minutes

Effective time used: 5 minutes

Notes: *not defined yet*

Task timeline



Code: 16:09:49 UTC, py, final, score: **100**

show code in pop-up

```python
def solution(A,K):
  N = len(A)
  # empty array case
  if not A: return A

  # if will loop more than once then make K the offset, full c
  if N<K:
    full_cycle_times = K//N
    K = K - (full_cycle_times*N)

  # non-cyclding returns original array
  if K==0 or N==K: return A

  # rearrange
  back_end = A[-K:]
  front_end = A[0:N-K]

  return back_end + front_end
```

Analysis summary

The solution obtained perfect score.

Analysis

collapse all **Example tests**

▼ example ✓ OK

example test

1. 0.020 s **OK**

collapse all **Correctness tests**

▼ extreme_empty ✓ OK

empty array

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ single ✓ OK

one element, 0 <= K <= 5

1. 0.016 s **OK**
2. 0.016 s **OK**
3. 0.016 s **OK**

▼ double ✓ OK

two elements, K <= N

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ small1 ✓ OK

small functional tests, K < N

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ small2 ✓ OK

small functional tests, K >= N

1. 0.016 s **OK**
2. 0.016 s **OK**
3. 0.016 s **OK**

▼ small_random_all_rotations ✓ OK

small random sequence, all rotations, N = 15

1. 0.016 s **OK**
2. 0.016 s **OK**
3. 0.016 s **OK**
4. 0.016 s **OK**
5. 0.016 s **OK**
6. 0.016 s **OK**
7. 0.016 s **OK**
8. 0.016 s **OK**
9. 0.016 s **OK**
10. 0.016 s **OK**
11. 0.016 s **OK**
12. 0.016 s **OK**
13. 0.016 s **OK**
14. 0.016 s **OK**
15. 0.016 s **OK**

▼ medium_random ✓ OK

medium random sequence, N = 100

1. 0.016 s **OK**

2. 0.016 s **OK**

▼ maximal ✓ OK

maximal N and K

1. 0.016 s **OK**
2. 0.016 s **OK**
3. 0.016 s **OK**
4. 0.016 s **OK**

Training center