Codility_

# CodeCheck Report: trainingDM5FQR-QSZ

Test Name:

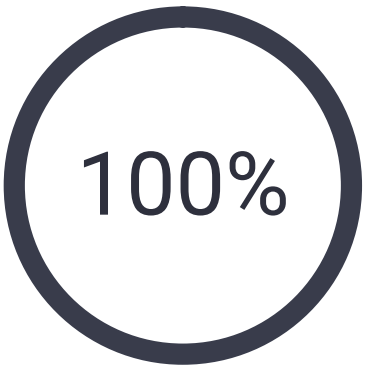
Check out Codility training tasks

Summary Timeline

## Tasks summary

| Task | Time spent | Score |
|---|---|---|
| **SocksLaundering** Python | 2 min | 100% |

## Total score




## Tasks Details

Medium

1. SocksLaundering

From drawers containing both clean and dirty socks, choose socks to launder in order to obtain the maximum number of clean pairs of socks.

| Task Score | Correctness | Performance |
|---|---|---|
| 100% | 100% | Not assessed |

### Task description

Bob is about to go on a trip. But first he needs to take care of his supply of socks. Each sock has its own color. Bob wants to take as many pairs of clean socks as possible (both socks in the pair should be of the same color).

Socks are divided into two drawers: clean and dirty socks. Bob has time for only one laundry and his washing machine can clean at most K socks. He wants to pick socks for laundering in such a way that after washing he will have a maximal number of clean, same-colored pairs of socks. It is possible that some socks cannot be paired with any other sock, because Bob may have lost some socks over the years.

### Solution

| | |
|---|---|
| Programming language used: | Python |
| Total time used: | 2 minutes |
| Effective time used: | 2 minutes |
| Notes: | *not defined yet* |

### Task timeline

Bob has exactly N clean and M dirty socks, which are described in arrays C and D, respectively. The colors of the socks are represented as integers (equal numbers representing identical colors).

For example, given four clean socks and five dirty socks:


If Bob's washing machine can clean at most K = 2 socks, then he can take a maximum of three pairs of clean socks. He can wash one red sock and one green sock, numbered 1 and 2 respectively. Then he will have two pairs of red socks and one pair of green socks.

Write a function:

```
def solution(K, C, D)
```

that, given an integer K (the number of socks that the washing machine can clean), two arrays C and D (containing the color representations of N clean and M dirty socks respectively), returns the maximum number of pairs of socks that Bob can take on the trip.

For example, given K = 2, C = [1, 2, 1, 1] and D = [1, 4, 3, 2, 4], the function should return 3, as explained above.

Assume that:

- K is an integer within the range [0..50];
- each element of arrays C and D is an integer within the range [1..50];
- C and D are not empty and each of them contains at most 50 elements.

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



13:51:39 13:53:03

Code: 13:53:03 UTC, py, final, score: **100** show code in pop-up

```
1  # you can write to stdout for debugging pu
2  # print("this is a debug message")
3
4  def solution(K, C, D):
5      # Implement your solution here
6      from collections import Counter
7
8      clean_counter = Counter(C)
9      dirty_counter = Counter(D)
10
11     pairs = 0
12
13     # Step 1: pair clean socks
14     for color, count in list(clean_counter
15         pair_count, odd = divmod(count, 2)
16         pairs += pair_count
17         if odd == 0:
18             del clean_counter[color]
19
20     # Step 2: pair clean with dirty socks
21     for color in list(clean_counter.keys()
22         if K > 0 and dirty_counter[color]
23             pairs += 1
24             K -= 1
25             dirty_counter[color] -= 1
26
27     # Step 3: pair dirty socks
28     for color, count in dirty_counter.item
29         pair_count, odd = divmod(count, 2)
30         possible_pairs = min(pair_count, K
31         pairs += possible_pairs
32         K -= possible_pairs * 2
33
34     return pairs
35
```

## Analysis summary

The solution obtained perfect score.

## Analysis

| expand all | Example tests | |
|---|---|---|
| ▶ | example<br>example test | ✔ OK |
| expand all | **Correctness tests** | |
| ▶ | smallest<br>smallest possible tests | ✔ OK |
| ▶ | small_simple<br>small simple tests, easy to solve even with heuristic approach | ✔ OK |
| ▶ | no_laundry<br>tests where K = 0 | ✔ OK |
| ▶ | odd_clean_with_odd_dirty<br>tests causing solution that's only | ✔ OK |

| | | |
|---|---|---|
| | pairing clean socks of odd count with dirty socks of odd count to fail | |
| ▶ | saving_one_used_dirty_check<br>tests causing solution that's not saving if dirty sock was used to pair with clean sock to fail | ✔ OK |
| ▶ | all_dirty<br>there are no clean socks taken | ✔ OK |
| ▶ | all_clean<br>there are no dirty socks taken after laundry | ✔ OK |
| ▶ | odd_even_dirty_check<br>odd and even number of dirty socks which do not match with clean socks | ✔ OK |
| ▶ | random_few_colors<br>randomly generated tests with only a couple of colors | ✔ OK |
| ▶ | maximal<br>maximal possible test cases | ✔ OK |