Codility_

CodeCheck Report: trainingBSU2YW-VE2

Test Name:

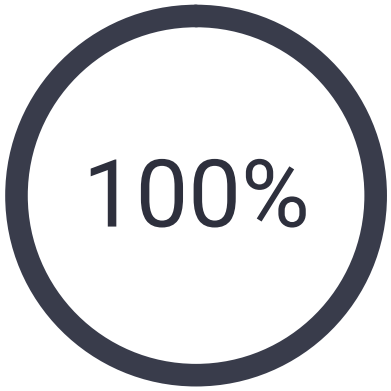
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
ParkingBill Python	7 min	100%

Total score



Tasks Details

1. ParkingBill

Elementary

Given two strings representing times of entry and exit from a car parking lot, find the cost of the ticket according to the given billing rules.

Task Score	Correctness	Performance
100%	100%	Not assessed

Task description

You parked your car in a parking lot and want to compute the total cost of the ticket. The billing rules are as follows:

- The entrance fee of the car parking lot is

Solution

Programming language used:	Python
Total time used:	7 minutes

2;
- The first full or partial hour costs 3;
- Each successive full or partial hour (after the first) costs 4.

You entered the car parking lot at time E and left at time L. In this task, times are represented as strings in the format `"HH:MM"` (where `"HH"` is a two-digit number between 0 and 23, which stands for hours, and `"MM"` is a two-digit number between 0 and 59, which stands for minutes).

Write a function:

```
def solution(E, L)
```

that, given strings E and L specifying points in time in the format `"HH:MM"`, returns the total cost of the parking bill from your entry at time E to your exit at time L. You can assume that E describes a time before L on the same day.

For example, given `"10:00"` and `"13:21"` your function should return 17, because the entrance fee equals 2, the first hour costs 3 and there are two more full hours and part of a further hour, so the total cost is 2 + 3 + (3 * 4) = 17. Given `"09:42"` and `"11:42"` your function should return 9, because the entrance fee equals 2, the first hour costs 3 and the second hour costs 4, so the total cost is 2 + 3 + 4 = 9.

Assume that:

- strings E and L follow the format "HH:MM" strictly;
- string E describes a time before L on the same day.

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

Effective time used: 7 minutes

Notes: *not defined yet*

Task timeline

14:16:32 14:23:06

Code: 14:23:05 UTC, py, final, score: **100** show code in pop-up

```
1  # you can write to stdout for debuggin
2  # print("this is a debug message")
3
4  def solution(E, L):
5      # Implement your solution here
6      entrance_fee = 2
7      first_hour_cost = 3
8      successive_hour_cost = 4
9
10     entry_hour, entry_minute = map(int
11     exit_hour, exit_minute = map(int,
12
13     # Calculate the duration of parkin
14     duration = (exit_hour - entry_hour
15
16     # Apply the billing rules
17     total_cost = entrance_fee + first_
18
19     # Subtract the first hour as it's
20     duration -= 60
21
22     if duration > 0:
23         # Calculate the number of addi
24         additional_hours = (duration +
25         total_cost += additional_hours
26
27     return total_cost
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	**Example tests**	
▶	example1 first example test	✔ OK
▶	example2 second example test	✔ OK
expand all	**Correctness tests**	
▶	under_ten_minutes	✔ OK

very short parking times, answer is always 5		
▶	random_under_hour short parking times, answer is always 5	✔ OK
▶	equal_hours_small parking for short time, always with complete hours	✔ OK
▶	random randomly generated test cases	✔ OK
▶	mixed medium and short intervals	✔ OK
▶	equal_hours_big long parking time, for complete hours	✔ OK
▶	random_big randomly generated parking times, for at least 20 hours	✔ OK
▶	maximum_result test cases giving maximum results or almost maximum results	✔ OK