← New question

NEW PROGRAMMING EXERCISE PYTHON 3

English Add translation

Title

Identify whether a string is a palindrome

Statement

Paragraph Source

Implement the function `string_is_palindrome(str)` so it returns `True` if the string is a palindrome and `False` if it is not a palindrome. The string will be at least one character long.

Note: the array always contains **at least one character**.

Settings

Domain
Python 3

Difficulty
Medium

Duration
01:00:00

Points
400

- [x] Allow the use of this question in automatically generated tests
- [ ] I authorize CoderPad to add this question to their question library

Execution environment

Running: Python 3 (check version details) Advanced settings

Initial candidate answer code

```
1   # Use print("messages...") to debug your solution.
2
3   def is_palindrome(test_string: str):
4
5       it_is_a_palindrome = False # Guilty until proven innocent
```

```
10
```

Initial candidate test code

```
# Add test code for the candidate.
# Use the print(...) function to output data.
# Only the lines of code between DISPLAY_BEGIN and DISPLAY_END
# will be shown to the candidate.

# To use the candidate's code, the Answer module must be imported
import Answer

# ##DISPLAY_BEGIN##
print(Answer.find_largest("madam"))          # True
print(Answer.find_largest("monsieur"))       # False
print(Answer.find_largest("kayak"))          # True
print(Answer.find_largest("true"))           # False
print(Answer.find_largest("false"))          # False
print(Answer.find_largest("two2owt"))        # True
print(Answer.find_largest("2357"))           # False
print(Answer.find_largest("638836"))         # True
print(Answer.find_largest("626"))            # True
print(Answer.find_largest("a"))              # True
print(Answer.find_largest("abcdefghijklm"))  # False
print(Answer.find_largest("fwgxkhkxgwf"))    # True
print(Answer.find_largest("iiii"))           # True
# ##DISPLAY_END##

```

Code validator

```
# The goal of the "Code validator" is to validate a programming task.
# A validator function will mark a candidate submission as "passed" if:
#   1) it does not throw an Exception (use assert(...) to throw Exceptions)
#   2) executes quickly enough
# otherwise the submission is marked as "failed".
#
# Each function declared here becomes a validator by filling its name in the
# tree of validators (see at the bottom of this page).
# If the parent validator fails, child validators will not be executed.
# Each validator function is started in its own process and should call the
# candidate's code using different test cases.

import Answer

_answer1 = True
_string1 = "madam"

_answer2 = False
_string2 = "madame"

_answer3 = True
_string3 = "273372"


def validate():
    assert(Answer.is_palindrome(_string1) == _answer1)


def validate_one_letter_string():
    assert(Answer.is_palindrome("a") == True)


def validate_french():
    assert(Answer.is_palindrome(_string2) == _answer2)


def validate_even_number_of_chars():
    assert(Answer.is_palindrome(_string3) == _answer3)

def validate_with_non_ascii():
    assert(Answer.is_palindrome("ABçðEF") == False)
```

Validator mapping (400 points in total)

Label	Method	Skill	Weight	Points	Status

Works with o	validate_one_	Reliability	1	35	
Works with n	validate_fren	Problem s(	3	110	
Works with a	validate_ever	Problem s(	3	110	
Works for a s	validate_with	Reliability	1	35	
Child validatc	Method nam		1	0	

Possible solution

```
1    def is_palindrome(test_string: str):
2        # This is a sample implementation to test your validators
3        return test_string[::-1] == test_string
4
```

SCREEN

INTERVIEW

CoderPad
SCREEN

M **Meta**

Tests

Questions

Candidates

Validation succeeded

Preview Save