Daniel Gamboa

danielgamboa10@gmail.com

BASIC INFO

Test: Computer Science - Python I - Module Project

Solved: 6/6

Similarity: none

Score: 1300/1300

Finished On: 05 Apr 2021

Time Taken: 82m/168h

Labels: -

Task	Solve Time	Score	Similarity
csStepsForUPERFramework	26sec	N/A	-
csUPERMostImportantActionInPlan	24sec	100/100	-
csAlphanumericRestriction	34min	300/300	none
csOppositeReverse	18min	300/300	none
csSquareAllDigits	7min	300/300	none
csRemoveTheVowels	7min	300/300	none

Task details: **csStepsForUPERFramework**

Description:

What are the four steps in the UPER problem-solving framework?

1. Understand
2. Plan
3. Execute
4. Reflect

Task details: **csUPERMostImportantActionInPlan**

Description:

What is the most important action to take during the Plan step of UPER?

- [x] Taking the problem description and transforming it into a complete, actionable plan to solve that problem (oftentimes using pseudocode to do so).

  (Correct)

- [ ] Developing a first-pass solution using actual working code.

  (Incorrect)

- [ ] Analyzing the time and space complexity of your solution and making sure it meets the provided benchmarks.

  (Incorrect)

- [ ] Asking lots of questions and clarifying your assumptions.

  (Incorrect)

Task details: **csAlphanumericRestriction**

Description:

Create a function that returns `True` if the given string has any of the following:

- Only letters and no numbers.
- Only numbers and no letters.

If a string has both numbers and letters or contains characters that don't fit into any category, return `False`.

Examples:

- `csAlphanumericRestriction("Bold") → True`
- `csAlphanumericRestriction("123454321") → True`
- `csAlphanumericRestriction("H3LL0") → False`

Notes:

- Any string that contains spaces or is empty should return `False`.

Solution (main.py3):

```
# Write a function that takes in a string and returns
# TRUE if all characters in that string are letters or
# if all characters in that string are numbers, but
# otherwise returns FALSE

def csAlphanumericRestriction(input_str):
    return input_str.isalpha() or input_str.isdecimal()

```

Task details: **csOppositeReverse**

Description:

Write a function that takes a string as input and returns that string in reverse order, with the opposite casing for each character within the string.

Examples:

- `csOppositeReverse("Hello World") → "DLROw OLLEh"`
- `csOppositeReverse("ReVeRsE") → "eSrEvEr"`
- `csOppositeReverse("Radar") → "RADAr"`

Notes:

- The input string will only contain alpha characters.

Solution (main.py3):

```
def csOppositeReverse(txt):
    rev = ""
    for i in range(len(txt)-1, -1, -1):
        rev += txt[i].swapcase()

    return rev

```

Task details: **csSquareAllDigits**

Description:

Create a function that given an integer, returns an integer where every digit in the input integer is squared.

Examples:

- `csSquareAllDigits(9119) -> 811181` because 9^2 = 81, 1^2 = 1, 1^2 = 1, and 9^2 = 81
- `csSquareAllDigits(2483) -> 416649` because 2^2 = 4, 4^2 = 16, 8^2 = 64, and 3^2 = 9

Solution (main.py3):

```
def csSquareAllDigits(n):
    store = ""
    for i in str(n):
        store += str(int(i)**2)

    return int(store)

```

Task details: **csRemoveTheVowels**

Description:

Given a string, return a new string with all the vowels removed.

Examples:

- csRemoveTheVowels("Lambda School is awesome!") -> "Lmbd Schl s wsm!"

Notes:

- For this challenge, "y" is not considered a vowel.

Solution (main.py3):

```
def csRemoveTheVowels(input_str):
    store = ""
    for c in input_str:
        if c != "a" and c != "e" and c != "i" and c != "o" and c != "u" \
        and c != "A" and c != "E" and c != "I" and c != "O" and c != "U":
            store += c

    return store


```