




Chad Diaz


chad-diaz@lambdastudents.com


BASIC INFO


< > Test: Computer Science - Python I  Solved: 5/6
- Module Project

 Similarity: none

 Score: 1200/1300

 Finished On: 06 May 2021

 Time Taken: 134m/168h

 Labels: -

Task	Solve Time	Score	Similarity
csStepsForUPERFramework	1min	N/A	-
csUPERMostImportantActionInPlan	2min	0/100	-
csAlphanumericRestriction	64min	300/300	none
csOppositeReverse	13min	300/300	none
csSquareAllDigits	31min	300/300	none
csRemoveTheVowels	23min	300/300	none

Task details: [csStepsForUPERFramework](#)

Description:

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

U = understand

P = plan

E= execute

R = review

Task details: **csUPERMostImportantActionInPlan**

Description:

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

- ☐ 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):

```
1 def csAlphanumericRestriction(input_str):
2     if input_str.isalpha():
3         return True
4
5     if input_str.isdigit():
6         return True
7
8     else:
9         return False
10
```

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")` → "DLR0w 0LLEh"
- `csOppositeReverse("ReVeRsE")` → "eSrEvEr"
- `csOppositeReverse("Radar")` → "RADAr"

Notes:

- The input string will only contain alpha characters.

Solution (main.py3):

```
1 def csOppositeReverse(txt):
2     txt = txt[::-1].swapcase()
3     return txt
4
5 csOppositeReverse("chaD AnD SARah ARE Awesome ParTners")
```

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):

```
1 def csSquareAllDigits(n):
2     string = ""
3
4     res = [int(x) for x in str(n)]
5
6     for number in res:
7         squared = (number ** 2)
8         newString = str(squared)
9         string += newString
10
11     return int(string)
12
13 csSquareAllDigits(654654)
14
15
```

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):

```
1 def csRemoveTheVowels(input_str):
2     newstr = input_str
3     vowels = ('a', 'e', 'i', 'o', 'u', 'I', 'O', 'U', 'A', 'E')
4     for x in input_str:
5         if x in vowels:
6             newstr = newstr.replace(x, "")
7     return newstr
8
9
```