CPE4040_Homework_1

September 4, 2024

CPE 4040 Spring 2024: Homework 2

1. Write your answer in the cell provided under each question.
2. **Write comments** in the code to explain your thoughts.
3. Show your execution result.
4. Do not use pandas, NumPy or other packages we have nor covered yet.
5. **Do your own work.**

0.0.1 Submission:

- **Submit this notebook file and the pdf version** - remember to add your name in the filename.
- Deadline: 11:59 pm, 2/4 (Sunday)

0.0.2 Q1: Mary Had a Little Lamb (15 Points)

[ ]:
```
title = 'mary HAD a little lamb'

lyrics = '''
Mary had a little limb,
Little limb, little limb.
Mary had a little limb
Its fleece was white as snow.

And everywhere that Mary went,
Mary went, Mary went,
Everywhere that Mary went
The limb was sure to go.

It followed her to school one day,
School one day, school one day.
It followed her to school one day
Which was against the rules.

It made the children laugh and play,
Laugh and play, laugh and play.
It made the children laugh and play,
To see a limb at school.
'''
```

0.0.3 Q1.1 The song title is not properly capitalized. Use string method to correct it to "Mary Had a Little Lamb".

[ ]:
```
title = title.title()
print(title)
```

```
Mary Had A Little Lamb
```

0.0.4 Q1.2 The word "lamb" is misspelled in several places. Please find them and correct the spelling.

[ ]:
```
lyrics = lyrics.replace('limb', 'lamb')
print(lyrics)
```

```
Mary had a little lamb,
Little lamb, little lamb.
Mary had a little lamb
Its fleece was white as snow.

And everywhere that Mary went,
Mary went, Mary went,
Everywhere that Mary went
The lamb was sure to go.

It followed her to school one day,
School one day, school one day.
It followed her to school one day
Which was against the rules.

It made the children laugh and play,
Laugh and play, laugh and play.
It made the children laugh and play,
To see a lamb at school.
```

0.0.5 Q1.3 How many words are in this song, not including commas, periods, spaces and new lines?

Note: Use string method split( ) first.

[ ]:
```
punc_remove = lyrics.replace(',', '').replace('.', '').replace('\n\n', ' ').
 ↪replace('\n', ' ').strip()
word_array = punc_remove.split(' ')
count = 0
for word in word_array:
    count += 1

print(count)
```

90

0.0.6 Q2: Palindrome or Symmetrical String (20 Points)

Write a Python program to check whether an input string is palindrome or symmetrical.

```
A sequence is said to be palindrome if one half of the string is the reverse of the other half
A sequence is said to be symmetrical if both halves of the string are the same, e.g., "abcabc"

**Note: if there are odd number of letters in a string, the middle one is ignored.
**Note: the string is case insensitive
```

Hint: use the string slice( ) method to divide the string into 2 halves.

To show your work, use those input strings: “level”, “Cosco”, “aaaAAaaa”.

[ ]:
```
def Palindrom(word):
    length = len(word) - 1
    for i in range(0, length):
        if (word[i] != word[length - i]):
            return False
    return True

def Symmetrical(word):
    length = len(word)
    offset = int(length / 2) + (length % 2)
    for i in range(0, offset - 1):
        if (word[i] != word[i + offset]):
            return False
    return True

print(Palindrom('level') or Symmetrical('level'))
print(Palindrom('Cosco') or Symmetrical('Cosco'))
print(Palindrom('aaaAAaaa') or Symmetrical('aaaAAaaa'))
```

```
True
False
True
```

0.0.7 Q3: List Comprehension - Exploring State Names (30 Points)

Given the a list of 50 state names, please use list comprehension to answer the following questions.

[ ]:
```
#-------------------------------------------------------------
# 50 state names as a list
#-------------------------------------------------------------
states = ['Alabama','Alaska','Arizona','Arkansas','California','Colorado',
          'Connecticut','Delaware','Florida','Georgia','Hawaii','Idaho',
          'Illinois','Indiana','Iowa','Kansas','Kentucky','Louisiana',
          'Maine','Maryland','Massachusetts','Michigan','Minnesota',
```

```
          'Mississippi','Missouri','Montana','Nebraska','Nevada',
          'New Hampshire','New Jersey','New Mexico','New York',
          'North Carolina','North Dakota','Ohio','Oklahoma','Oregon',
          'Pennsylvania','Rhode Island','South Carolina','South Dakota',
          'Tennessee','Texas','Utah','Vermont','Virginia','Washington',
          'West Virginia','Wisconsin','Wyoming']
```

0.0.8 Q3.1 What are the top 3 states with the longest one-word name?

Hint: Remember do not consider states such New York, North Carolina etc. which have two words.

[ ]:
```
one = {
    "state": "",
    "length": 0
}
two = {
    "state": "",
    "length": 0
}
three = {
    "state": "",
    "length": 0
}
for state in states:
    if (len(state.split(' ')) > 1):
        pass
    else:
        if (len(state) > one["length"]):
            one["length"] = len(state)
            one["state"] = state
        elif (len(state) > two["length"]):
            two["length"] = len(state)
            two["state"] = state
        elif (len(state) > three["length"]):
            three["length"] = len(state)
            three["state"] = state

print(one["state"])
print(two["state"])
print(three["state"])
```

```
Massachusetts
Pennsylvania
Washington
```

0.0.9 Q3.2 Which states start and end with the same letter, for example, Alaska starts and ends with 'a'?

[ ]:
```
for state in states:
    if (state[0].lower() == state[len(state) - 1]):
        print(state)
```

```
Alabama
Alaska
Arizona
Ohio
```

0.0.10 Q3.3 Which alphabetic letter is not present in any of the state names? Yep, there's only one.

[ ]:
```
import string
alphabet = list(string.ascii_lowercase)

for state in states:
    for letter in state:
        for i in alphabet:
            if (letter.lower() == i):
                alphabet.remove(i)

print(alphabet)
```

```
['q']
```

0.0.11 Q3.4 Which states names have no character repetition? For example, Florida has 7 distict letters and no repetition.

[ ]:
```
dup = False
for state in states:
    dup = False
    for i in range(len(state)):
        for j in range(i):
            if state[i].lower() == state[j].lower():
                dup = True
                break
    if (dup == False):
        print(state)
```

```
Florida
Idaho
Iowa
Maine
New York
Texas
Utah
```

```
Vermont
Wyoming
```

0.0.12 Q3.5: Which state names do not have the letter 'a' at all?

[ ]:
```
for state in states:
    if not ('a' in state.lower()):
        print(state)
```

```
Connecticut
Illinois
Kentucky
Mississippi
Missouri
New Jersey
New Mexico
New York
Ohio
Oregon
Tennessee
Vermont
Wisconsin
Wyoming
```

Q4: List Operations (20 points)

You will first generate a list of 10 random integers from 1 to 1000 by using the Python random module. Save the list (let's call it A) and write a short snippet for each of the following task:

1. Pick and sort first the odd numbers then the even numbers in ascending order and print the final list, e.g. [1 3 51 4 16 32]
2. Extract only the items that are less than 250.
3. Create a new list where each item y is the square of the correpsonding item in A.
4. Extract the items that are NOT divisible by 3 and NOT divisible by 7

[ ]:
```
import random

# Note: you can use random.randint() to generate the random integers
list = [random.randint(1, 1000) for i in range(10)]
print(list)
```

```
[727, 924, 311, 337, 872, 795, 696, 239, 235, 985]
```

[ ]:
```
#Q4_1
even = [x for x in list if not x % 2]
odd = [x for x in list if x % 2]
even.sort()
odd.sort()

for i in even:
```

```
    odd.append(i)
print(odd)
```

```
[235, 239, 311, 337, 727, 795, 985, 696, 872, 924]
```

[ ]:
```
#Q4_2
less = [x for x in list if x < 250]
print(less)
```

```
[239, 235]
```

[ ]:
```
#Q4_3
square = [y**2 for y in list]
print(square)
```

```
[528529, 853776, 96721, 113569, 760384, 632025, 484416, 57121, 55225, 970225]
```

[ ]:
```
#Q4_4
new_list = [x for x in list if not (x % 3 == 0 and x % 7 == 0)]
print(new_list)
```

```
[727, 311, 337, 872, 795, 696, 239, 235, 985]
```

0.0.13 Q5: Order of Planets (15 Points)

In the solar system, the planets, in order of their distances from the Sun (closest to farthest), are Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune. Given an incomplete and out-of-order list, please use list methods to make it right.

[ ]:
```
planets = ['venus', 'mercury', 'mars', 'jupiter', 'earth']
```

[ ]:
```
planets.append('saturn')
planets.append('uranus')
planets.append('nepture')
planets.insert(planets.index('venus'), planets.pop(planets.index('mercury')))
planets.insert(planets.index('mars'), planets.pop(planets.index('earth')))
print(planets)
```

```
['mercury', 'venus', 'earth', 'mars', 'jupiter', 'saturn', 'uranus', 'nepture']
```