10/10

Group_08_Exercise_02

December 7, 2020

1 Exercice 02:

The following exercice requires some understanding in the following subjects: - understand conditions in python - write some loops

1.1 Loop:

1.1.1 1. Use for, .split(), and if to create a Statement that will print out words that start with 's':

[1]:
```
# it should print: "start", "s", "simple", "sentence"

# change the string in list by a space separation by default with split

# print all the words starting with a "s"
#sentence = input("please type a sample sentence:")

sentence =  "start anything with s may be a simple sentence"

for word in sentence.split():
    if word[0] == 's':
        print(word)
```

```
start
s
simple
sentence
```

1.1.2 2. Given a list l_list, delete all the 0 values in the list l_list. (hint: see the methods for list in the python documentation)

[2]:
```
# here is the list l_list
l_list = [ 2 , 5 , 8 , 0 , 0 , 0 , 0 , 0 , 5 , 2 , 3 , 9 , 1 , 5 , 3 , 0, 11,␣
 ↪13, 0, 5]
# WARNING, you cannot delete an element in a list during a loop, why? Because␣
 ↪of shrinking list
# the best way is to create another list and save all the wanted value in this␣
 ↪list.
# let's create another empty list to save the value without zero
```

```
new_list = []

for i in l_list:
     if i != 0:
            new_list.append(i)

print (new_list)
```

```
[2, 5, 8, 5, 2, 3, 9, 1, 5, 3, 11, 13, 5]
```

1.1.3 3. We consider the following dictionary (students) whose keys are the names of the students and the values of the keys are the overall averages obtained by passing the final exam.

Write a Python program that partitions this dictionary into two sub-dictionaries:

1. admittedStudents whose keys are the admitted students and the values of the keys are the averages obtained (average greater than or equal to 10).
2. nonAdmittedStudent whose keys are the non-admitted students and whose key values are the averages obtained (average less than or equal to 10).

[3]:
```
students = {"student_1" : 13 , "student_2" : 17 , "student_3" : 9 , "student_4"␣
 ↪: 15 ,
                         "student_5" : 8 , "student_6" : 14 , "student_7" : 16␣
 ↪, "student_8" : 12 ,
                         "student_9" : 13 , "student_10" : 15 , "student_11" :␣
 ↪14 , "student_112" : 9 ,
                         "student_13" : 10 , "student_14" : 12 , "student_15" :␣
 ↪13 , "student_16" : 7 ,
                         "student_17" : 12 , "student_18" : 15 , "student_19" :␣
 ↪9 , "student_20" : 17}

# your code here
# from what i understend , The average greater than or equal to 10 should be in␣
 ↪admittedStudents dictionary.
# The average less than or equal to 10 should be in nonAdmittedStudent␣
 ↪dictionary. This leaves student_13 in both dictionaries
# The solution is as follows

admittedStudents  = {}
nonAdmittedStudent = {}

for k, v in students.items():
    if ((v >= 10) and (v <= 10)):
        admittedStudents[k]  = v
        nonAdmittedStudent[k]  = v
    elif (v <= 10):
        nonAdmittedStudent[k] = v
    elif (v >= 10):
         admittedStudents[k] = v


print("Admitted Students:\n", admittedStudents )
print("\nNon Admitted Students:\n", nonAdmittedStudent )
```

```
Admitted Students:
 {'student_1': 13, 'student_2': 17, 'student_4': 15, 'student_6': 14,
'student_7': 16, 'student_8': 12, 'student_9': 13, 'student_10': 15,
'student_11': 14, 'student_13': 10, 'student_14': 12, 'student_15': 13,
'student_17': 12, 'student_18': 15, 'student_20': 17}

Non Admitted Students:
 {'student_3': 9, 'student_5': 8, 'student_112': 9, 'student_13': 10,
'student_16': 7, 'student_19': 9}
```

1.1.4 4. Use for loop to print an isosceles triangle with the character * . The user should be asked for a number and the the program print the triangle with the *character.

[4]:
```
# if for example, the user type 3, the program should print this:
#         *
#        ***
#       *****
#
#
#  if the user typed 5, the program should print this:
#         *
#        ***
#       *****
#      *******
#     *********
# and so on...

space = 2 * int(input("please enter the number of lines you want print\n"))

for star in range(1, space , 2):
    print(" " * space + star * "*")
    space = space-1

    # The output does not seem to be a perfect isosceles triangle optically.␣
 ↪But, when the same code is executed in spyder environment
    # But, when the same code is executed in spyder environment. The output␣
 ↪renders a optically good isosceles triangle.
```

```
please enter the number of lines you want print
 5

         *
        ***
       *****
      *******
     *********
```

1.1.5 5. A wolf in sheep's clothing

Wolves have been reintroduced to Great Britain. You are a sheep farmer, and are now plagued by wolves which pretend to be sheep. Fortunately, you are good at spotting them.

Warn the sheep in front of the wolf that it is about to be eaten. Remember that you are standing at the front of the queue which is at the end of the array:

```
[sheep, sheep, sheep, sheep, sheep, wolf, sheep, sheep]      (YOU ARE HERE AT THE FRONT OF THE
   7      6      5      4      3             2      1
```

If the wolf is the closest animal to you, return "Pls go away and stop eating my sheep". Otherwise, return "Oi! Sheep number N! You are about to be eaten by a wolf!" where N is the sheep's position in the queue.

Note: there will always be exactly one wolf in the array.

Examples:

```
sheep_queue_0 = ["sheep", "sheep", "sheep", "wolf", "sheep"]
# should print
-> 'Oi! Sheep number 1! You are about to be eaten by a wolf!'

sheep_queue_1 = ['sheep', 'sheep', 'wolf']
# should print
-> 'Pls go away and stop eating my sheep'
```

[7]:
```python
sheep_queue_0 = ["sheep", "sheep", "sheep", "wolf", "sheep"]  # test your code␣
 ↪with sheep_queue_0 and sheep_queue_1 also
# your code here
print("\nSpotting the Wolf in Sheep Queue 0:")
if sheep_queue_0[-1] == 'wolf':
    print('Pls go away and stop eating my sheep')
else:
    for i , j in enumerate(sheep_queue_0):
        k = abs(len(sheep_queue_0) - i)
        if j == 'wolf':
            print(f'Oi! Sheep number {k-1}! You are about to be eaten by a wolf!
 ↪')
 ###################################################################
sheep_queue_1 = ['sheep', 'sheep', 'wolf']
print("\nSpotting the Wolf in Sheep Queue 1:")
```

```
if sheep_queue_1[-1] == 'wolf':
    print('Pls go away and stop eating my sheep')
else:
    for i , j in enumerate(sheep_queue_1):
        k = abs(len(sheep_queue_1) - i)
        if j == 'wolf':
            print(f'Oi! Sheep number {k-1}! You are about to be eaten by a wolf!
 ↪')

####################################################################
sheep_queue_2 = ['wolf', 'sheep', 'sheep', 'sheep', 'sheep', 'sheep', 'sheep']
print("\nSpotting the Wolf in Sheep Queue 2:")


if sheep_queue_2[-1] == 'wolf':
    print('Pls go away and stop eating my sheep')
else:
    for i , j in enumerate(sheep_queue_2):
        k = abs(len(sheep_queue_2) - i)
        if j == 'wolf':
            print(f'Oi! Sheep number {k-1}! You are about to be eaten by a wolf!
 ↪')
```

```
Spotting the Wolf in Sheep Queue 0:
Oi! Sheep number 1! You are about to be eaten by a wolf!

Spotting the Wolf in Sheep Queue 1:
Pls go away and stop eating my sheep

Spotting the Wolf in Sheep Queue 2:
Oi! Sheep number 6! You are about to be eaten by a wolf!
```

1.1.6 6. time(s)

You receive a string, and you need to return a **string** that shows how many times each letter shows up in the string by using the sign plus "+"

For example:

"Chicago" –> "c:++,h:+,i:+,a:+,g:+,o:+" As you can see, the letter c is shown only once, but with 2 pluses.

The return string should include only the letters (not the dashes, spaces, apostrophes, etc). There should be no spaces in the output, and the different letters are separated by a comma (,) as seen in the example above.

Note that the return string must list the letters in order of their first appearence in the original string.

More examples:

"Bangkok" –> "b:+,a:+,n:+,g:+,k:++,o:+"

"Las Vegas" –> "l:+,a:++,s:++,v:+,e:+,g:+"

[6]:
```python
# your code here. Test it with the 3 words above
s = str(input("Please type a String:\n"))
st = "".join(s.lower().split())
unique = {}
result_str = ""

for i in range(len(st)):
    unique[st[i]] = st.count(st[i])

for k,v in unique.items():
    result_str = result_str + k + ':' + v * '+' + ','

result_str = result_str[:-1]


print(result_str)
```

```
Please type a String:
 chicago

c:++,h:+,i:+,a:+,g:+,o:+
```

[8]:
```python
s = str(input("Please type a String:\n"))
st = "".join(s.lower().split())
unique = {}
result_str = ""

for i in range(len(st)):
    unique[st[i]] = st.count(st[i])

for k,v in unique.items():
    result_str = result_str + k + ':' + v * '+' + ','

result_str = result_str[:-1]


print(result_str)
```

```
Please type a String:
 Bangkok

b:+,a:+,n:+,g:+,k:++,o:+
[9]: s = str(input("Please type a String:\n"))
     st = "".join(s.lower().split())
     unique = {}
     result_str = ""

     for i in range(len(st)):
         unique[st[i]] = st.count(st[i])

     for k,v in unique.items():
         result_str = result_str + k + ':' + v * '+' + ','

     result_str = result_str[:-1]


     print(result_str)
```

```
Please type a String:
 Las Vegas

l:+,a:++,s:++,v:+,e:+,g:+
```