Assignment - Python Data Types and Functions

Submission:

- Submit pdf and html files of this notebook to Canvas. To create a pdf file: to to File menu, Print Preview and print your notebook as a pdf file.
- Each question is worth 5 points.

[Tip]: To execute the Python code in the code cell below, click on the cell to select it and press `Shift` + `Enter`.

[Tip]: Save your notebook frequently, click on the save button or press `Ctr` + `S`.

Question 1

Define a function **count_int** in Python that will take a list of any size as an input and return the number of elements of type *int* in that list.

For example, the *list_one* has 4 elements that are integers and your function should return 4.

In [1]:

```
# your code here - provide the definition of the function
def count_int (some_list):

    count = 0
    for ind,val in enumerate(some_list):
        if type(val) == int:
            count += 1

    print(count)
    return count # the number of int element
```

In [2]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
list_one = ['a', 3, 'abc', 3.4, 5, 7,7,(3,3,3)]
list_two = [('abc'), 6.0, [3,4,4], 'number three', [4, 4.5]]
list_three = [('abc'), 6.0, 3,4,4, 'number three', [4, 4.5]]
```

In [3]:

```
# DO NOT CHANGE THIS CODE
assert(count_int(list_one) == 4)
assert(count_int(list_two) == 0)
assert(count_int(list_three) == 3)
```

```
4
0
3
```

Question 2

Define a function **odd_even** in Python that will take a list of any size as an input and return a string "odd" if the list includes an odd number of elements. Otherwise, the function will return a string "even". For example, the *list_one* has 4 elements that are integers and your function should return 4.

In [39]:

```
# your code here - provide the definition of the function
def odd_even (some_list):
    # ... your code here

    if len(some_list) %2 == 1:
        print('odd')
        return 'odd'

    else:
        print('even')
        return 'even'
```

In [40]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
list_one = ['a', 3, 'abc', 3.4, 5, 7,7,(3,3,3)] # this returns "even"
list_two = [('abc'), 6.0, [3,4,4], 'number three', [4, 4.5]]
list_three = [('abc'), 6.0, 3,4,4, 'number three', [4, 4.5]]
```

In [41]:

```
# DO NOT CHANGE THIS CODE
assert (odd_even(list_one) == 'even')
assert (odd_even(list_two)== 'odd')
assert (odd_even(list_three) == 'odd')
```

```
even
odd
odd
```

Question 3

Define a function **total_char** in Python that will take a list of any size as an input and return the total number of characters in that list.

For example, the **list_one = ['a', '3', 'abc', 'd', 'f', '7','a',('3','3','a')]** includes 12 characters.

- For simplicity, your function will take a list with at most one nested index, such as **list_one[i][i]**.
- A list will contain a string or a list of strings or a tuple of strings
- Test your function with example lists

In [61]:

```
# your code here - provide the definition of the function
def total_char(some_list):
    count = 0
    for thing in some_list:
        if type(thing) == str:

            count+=len(thing)

        elif type(thing) == tuple or type(thing) == list:
            count+=len(thing)
    return count
```

In [62]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
list_one = ['a', '3', 'abc', 'd', 'f', '7','a',('3','3','a')]
list_two = [('abc'), 'a', ['3','4','4'], 'number three', ['4', 'a']]
list_three = [('abc'), 'apples', ['4', 'a']]
a = total_char(list_one)
print(a)
b = total_char(list_two)
print(b)
c = total_char(list_three)
print(c)
```

```
12
21
11
```

In [63]:

```
# DO NOT CHANGE THIS CODE
assert(total_char(list_one) == 12)
assert(total_char(list_two) == 21)
assert(total_char(list_three) == 11)
```

Question 4

Define a function **count_a** in Python that will take a list of any size as an input and return the total number the characters 'a' appears in that list. For example, the *list_one* has four characters 'a' and your function should return 4.

- For simplicity, your function will take a list with at most one nested index, such as **list_one[i][i]**.
- An input to the **count_a** function is a list that contains a character or a list of character or a tuple of characters.
- Test your function with example lists

In [49]:

```
# your code here - provide the definition of the function
def count_a (some_list, char_to_find = 'a'):

    count = 0
    for x in some_list:
        for y in x:
            if y == char_to_find:
                count+=1


    print(count)
    return count
```

In [50]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
list_one = ['a', '3', 'a', 'd', 'f', '7','a',('3','3','a')] # 4
list_two = [('aba'), 'a', ['3','4','4'], 'number three', ['4', 'a']]
list_three = ['a', 'a','a' ,['a','b','a']]
```

In [51]:

```
# DO NOT CHANGE THIS CODE
assert(count_a(list_one) == 4)
assert(count_a(list_two) == 4)
assert(count_a(list_three) == 5)
```

```
4
4
5
```

Question 5

Define a function **longest_list** in Python that will take a list as in input and return a new list.

- For simplicity, your function will take a list with at most one nested index, such as **list_one[i][i]**.
- Each element in an input list is a list of integers.
- Your function will return the longest list of integers.
- For instance longest_list(one) will return a new list with the following elements [2,2,2,5]

In [52]:

```
# your code here - provide the definition of the function
def longest_list (some_list):
    print(max(some_list, key=len))
    return max(some_list, key=len)
```

In [53]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
one = [[4], [2,3], [2,2,2,5], [5]]
two = [[2,2,2,2,2,2,2,], [2,3], [2,2,2], [33,1]]
three = [[2,1,3], [2,3], [3,2,6,1], [7]]
```

In [54]:

```
# DO NOT CHANGE THIS CELL
assert(longest_list(one) ==[2, 2, 2, 5] )
assert(longest_list(two) ==[2, 2, 2, 2, 2, 2] )
assert(longest_list(three) ==[3, 2, 6, 1] )
```

```
[2, 2, 2, 5]
[2, 2, 2, 2, 2, 2]
[3, 2, 6, 1]
```

Question 6

Create a dictionary `ABC` that includes the following set of key and values:

- key: "a" values: "apple", "ant", "astute", “android”
- key: “b” values: “bankruptcy”, “baby”, “blue”, “big”
- key: “c” values: “cats”, “calm”, “car”, “candy”,

Keys are the strings and values are lists.

In [55]:

```
ABC = {
        'a': ['apple', 'ant', 'astute', 'android'],
        'b': ['bankruptcy', 'baby', 'blue', 'big'],
        'c': ['cats', 'calm', 'car', 'candy']
}
```

Write the function `replace_me(old_word, new_word)` that will find the word in a dictionary and replace it with a new one.

For instance : `replace_me(“big”,"small")` will replace “big” with the “small”.

In [56]:

```
ABC = {
        'a': ['apple', 'ant', 'astute', 'android'],
        'b': ['bankruptcy', 'baby', 'blue', 'big'],
        'c': ['cats', 'calm', 'car', 'candy']
}

def replace_me(old_word, new_word):
    for key, value in ABC.items():
            if isinstance(value, list):
                for i in range(len(value)):
                    if isinstance(value[i], str):
                        if old_word in value[i]:
                            value[i] = value[i].replace(old_word, new_word)
```

In [57]:

```
# DO NOT CHANGE THIS CELL
print(replace_me("cats" ,"dogs"))
print(replace_me("blue","yellow"))
print(ABC)
```

```
None
None
{'a': ['apple', 'ant', 'astute', 'android'], 'b': ['bankruptcy', 'baby', 'yellow', 'big'], 'c': ['dogs', 'calm', 'car', 'candy']}
```

In [86]:

```
# answer
```

Bonus Question 7 - 3 points

- Define a function **multiply** in Python that will take a list of any size as an input.
- Each element of an input list may contain a **list of integers** ONLY.
- For simplicity, your function will take a list with at most one nested index, such as **list_one[i][i]**.
- Use examples below to test your function.
- The **multiply** function will return a list of **integers**, the length of this list is equal to the length of the original list.
- Each item in a new returned list is an integer that is the multiplication result of all integers in each item.
- For example, **multiply([[4], [2,3], [2,2,2], [7]])** will return a new list with the following elements **[4,6,8,7]**.

In [58]:

```
# your code here - provide the definition of the function
def multiply (some_list):
    a_list = []
    for listt in some_list:
        result = 1
        for element in listt:
            result *= element
        a_list.append(result)
    print(a_list)
    return a_list # the number of int element
```

In [59]:

```
# test your function with the following input- DO NOT CHANGE THIS CELL
mult_one = [[4], [2,3], [2,2,2], [5]]
mult_two = [[2], [2,3], [2,2,2], [33,1]]
mult_three = [[2,1], [2,3], [3,2,6], [7]]
```

In [60]:

```
assert (multiply(mult_one) == [4, 6, 8, 5])
assert (multiply(mult_two) == [2, 6, 8, 33])
assert (multiply(mult_three) == [2, 6, 36, 7])
```

```
[4, 6, 8, 5]
[2, 6, 8, 33]
[2, 6, 36, 7]
```

The end!

Submit your notebook in a pdf and html format to Canvas