Lab 1 : Recursion

Instruction:

1) Complete the assignment and name each file as **LabxPy_studentIDDash.py**, where x is a lab number and y is a problem number.
   For example, **Lab1P3_623040000-0.py**.
   A wrong file name will not be graded.
2) Late submission will be penalized 50% of the lab score.
3) One (1) late submission is allowed per semester without a penalty.
4) Turn in your file in the google classroom. A due date will be specified in the classroom.

---

1. Write a Python program to calculate a factorial of an input n by using recursion.

Output example:

```
Enter n: 5
n! is 120
```

2. Write a Python program to reverse values in a given list of size n.
   a. Get n as an input
   b. Get n integers from a console and into a list
   c. Display an original list
   d. Make values in the list reverse by using recursion, *no pre-defined method/function is allowed.*
   e. Display a current list if there is any change.

Output example:

```
Enter n: 5
Enter 5 integers: 10 20 30 40 50
Original list: 10 20 30 40 50
Start reversing ...
50 20 30 40 10
50 40 30 20 10
50 40 30 20 10
Reversed list: 50 40 30 20 10
```

Note: You don’t need to have the same in-process sequences as I did.

3. Write a Python program to calculate the sum from 1 to x by using recursion. For example, if x = 10, calculate 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10.

   Hint: Think of the factorial recursive function.

Output example:

```
Enter an integer x: 10
1
3
6
10
15
21
28
36
45
55
Sum of 1 to 10 is 55
```

Note: You don’t need to have the in-process sequences as I did.

4. Write a Python program to determine how many digits a positive integer has by using recursion.

   Hint : if you keep dividing a number by 10 (integer division), until the number is less than 10, you can count the digits of the original number.

Output example:

```
Enter a positive integer: 2367514
count 1
count 2
count 3
count 4
count 5
count 6
count 7
2367514 has 7 digit(s)
```

Note: You don’t need to have the in-process sequences as I did.

5. check whether a given word is a palindrome, by using recursion. A palindrome is a word that is read the same from front to back and from back to front.

   For example, ahha is a palindrome while ahhaa is not.

   Use the following code template:

```
def isPalin(word):
    pass
```

```
myword = ""
myword = input("Enter a word: ")
while(myword.lower() != "done"):
    print(myword, isPalin(myword))
    myword = input("Enter a word: ")
```

Output example:

```
Enter a word: Civic
Civic True
Enter a word: level
level True
Enter a word: ahha
ahha True
Enter a word: ahhaa
ahhaa False
Enter a word: palindrome
palindrome False
Enter a word: a
a True
Enter a word: aa
aa True
Enter a word: ah
ah False
Enter a word: done
```