PROGRAMMING FOR ALL – SPRING 2021 - WEEK #6 – LAB – 210505

GENERAL LAB RULES:

1. **Every student has to join lab sessions.**
2. A goal of lab assignments is to practice topics covered in class during each week. Each lab has two parts: Theory & Programming.
3. In the theory part, you are expected to read pieces of codes and to predict outputs of these codes. You should not use Python for these predictions; you should test your skills to understand codes. When everyone is ready, TA displays correct outputs.
4. In the programming part, you are expected to make programs for assigned problems. You are expected to make all programs during the lab time, however, only one specified program should be submitted to CANVAS.
5. TA has to check your lab work for the full lab credit.

THIS WEEK TOPICS:

MORE ABOUT STRINGS – TESTING, SEARCHING, MANIPULATING

PART I. – THEORY

PROGRAM OUTPUTS

First predict and then verify the outputs of the following print statements. Don't use Python, open an empty file or use a paper to predict. TA will share correct answers with you. The goal of these exercises is to understand programming codes.

#1

```
my_string='TODAY'
s=len(my_string)
print(my_string[1:s-1])
```

#2

```
my_string='TODAY'
if 'd' in my_string:
    print('YES')
else:
    print('NO')
```

#3

```
string='TODAY is 20/05/06.'
print(string.isalnum())
print(string.isalpha())
print(string.isdigit())
print(string.islower())
print(string.isupper())
```

#4

```
string='TODAY is 20/05/06.'
count_up=0
count_dig=0
for char in string:
    if char.isupper():
        count_up+=1
    elif char.isdigit():
        count_dig+=1
print(count_up, count_dig)
```

#5

```
your_string='We like Python.'
print(your_string[:2].lower())
print(your_string.rstrip('.'))
print(your_string[-7:].upper())
```

#6

```
string='We like Python.'
print(string.endswith('n'))
print(string.find('like'))
print(string.find('.'))
print(string.find('a'))
print(string.replace('We','I'))
```

#7

```
text='abracadabra'
text_1=text[2:8].upper()
if 'B' in text_1:
    text_2=text_1*2
else:
    text_2=text_1*3
print(text_2)
```

#8

```
text='abracadabra'
position=text.find('r')
print(text[position:])
```

PART II. PROGRAMMING

Note: In all programs you can use only methods and tools already introduced in the class.

PROGRAM #1:

Problem: INTEGER TRANSFORMATION

Write a program that prompts a user to enter a positive integer and then displays a modified integer.

Modification rules:

- If in the entered integer is the digit 8, this digit is replaced by the symbol 'X' and all other digits stay as they are (X transformation) ;
- If in the entered integer is not the digit 8, then all odd digits are replaced by the symbol 'Y' and all other digits stay as they are (Y transformation).

The user can enter multiple inputs. If the user enters the symbol #, the program ends. The program also validates, whether the input is a positive integer (except of the ending symbol #).

At the end the program displays the number of X and Y transformations.

Example of a possible output:

```
This program displays a transformed form of a positive integer.
If you enter # for the integer, the program ends.
Enter a positive integer: 12358
The number was transformed to:  1235X

Enter the next positive integer: 114477
The number was transformed to:  YY44YY

Enter the next positive integer: yes
The input is not valid.
Enter a positive integer: ok
The input is not valid.
Enter a positive integer: 878787
The number was transformed to:  X7X7X7

Enter the next positive integer: #

The number of X transformations:  2
The number of Y transformations:  1
>>>
```

PROGRAM #2:

Problem: STRING PARTS– must be submitted to CANVAS as lab6.py.

Start the program with: # your name.

Write a program that prompts a user to enter two strings (as two inputs). Then the program displays:

- If the second entered string is in the first entered string, the program creates and displays portions of the first entered string preceding and following the second entered string.
  Example: If the first entered string is "ALPHABET" and the second is "HA", then the program displays "ALP" and "BET".

- If the second entered string is not in the first entered string, the program creates and displays one new string where the first entered string is followed by the second and they are separated by the symbol ' & '.
  Example: If the first entered string is "ALPHABET" and the second is "YES", then the program displays "ALPHABET & YES".

Hint: use the method `.find()` to locate the second string.

The user can enter multiple couples of strings, if the user enter for the first string "@@@" the program ends.

Example of a possible output:

```
This program displays a new strings created from two entered strings.
To end the program - enter for the first string @@@.

Enter the first string: EXCELLENT
Enter the second string: CEL
before:  EX after:  LENT

Enter the next first string: YESTERDAY
Enter the second string: ABC
together:  YESTERDAY & ABC

Enter the next first string: 12567845
Enter the second string: 567
before:  12 after:  845

Enter the next first string: @@@
>>>
```