PROGRAMMING FOR ALL – SPRING 2021 - WEEK #7 – LAB – 210512

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:

FUNCTIONS – INTRO, VOID, PARAMETER

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

```
def printing():
      for symbol in ['@','$','&']:
            for num in range(2,7,3):
                  print(symbol*num)
printing()
```

#2

```
def count_odd(string_in):
      sum_odd=0
      for char in string_in:
            if char.isdigit() and int(char)%2==1:
                  sum_odd+=int(char)
      print(sum_odd)
count_odd('N1L5XX4873')
```

#3

```
def yes_no(string):
      if string[1]!=string[-4]:
            print('NO')
      else:
            print('YES')
yes_no('R A C O O N')
```

#4

```
def validation(value):
      valid=False
      if value<=100 or value>=200:
            valid=True
      print(valid)
validation(150)
validation(250)
```

#5

```
def operation(string):
      if len(string)<=4:
            print(string[0])
      else:
            print(string[4])
operation('SUPPER')
```

#6

```
def info(temperature):
      while temperature<10:
            print('LOW',end=' ')
            temperature+=1
      print('OK')
info(6)
info(12)
```

#7

```
def squares(top):
      base=0
      while base**2<=top:
            print(base**2, end=' ')
            base+=1
      print('')
squares(12)
squares(25)
```

PART II. PROGRAMMING

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

PROGRAM #1: FUNCTIONS

- Write a function `factorial` which takes as an argument a nonnegative integer and displays the factorial of the entered integer.

  Test the function using values 0, and 15.

- Write a function `code` which takes as an argument a positive integer and then transforms the integer to a code and displays the code. Rules of transformation:
  - Each even digit in the integer is replaced by the symbol '*$*';
  - Each odd digit in the integer is replaced the symbol '#'.

  Test the function using values 10023, and 789456.

- Write a function `validation` which takes as an argument a string and displays whether the string has a valid form or no. A string is valid if:
  - it has at least six characters but not more than ten characters;
  - it has at least two symbols '$' or '&' (it also includes mixes of those symbols.)

  Test the function using the strings 12JK&5$11, and $$G&&.

PROGRAM #2:

Problem: CODES – must be submitted to CANVAS as lab7.py.

Start the program with: # your name.

STEP 1.

Write a function `code` which takes as an argument a positive integer and then transforms the integer to a code and displays the code. Rules of transformation:

- Each even digit in the integer is replaced by the symbol '*$*';
- Each odd digit in the integer is replaced the symbol '#'.

Note: this should be done in the program #1.

STEP 2.

Write a program which is using the function `code` such way that a user can enter a positive integer repeatedly and the program displays for each entered integer the created code.

By entering a non-positive integer (includes 0) the user ends the program.

At the end the program displays the number of positive integers entered by the user.

See a possible output:

```
This program transforms positive integers to codes.
By entering a non-positive integer you end the program.

Enter a positive integer: 1458955
The code is:  #*X*#*X*###

Enter the next positive integer: 4444556
The code is:  *X**X**X**X*##*X*

Enter the next positive integer: 89562233
The code is:  *X*##*X**X**X*##

Enter the next positive integer: -5

You entered 3 positive integers.
Thank you for using my program!
>>>
```