# PROGRAMMING FOR ALL – SPRING 2021 - WEEK #8 – LAB – 210519

## 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.

## TOPICS:

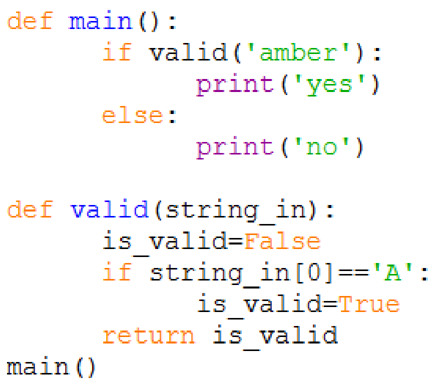
FUNCTIONS – VALUE RETURNING, RANDOM NUMBERS

## 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.**

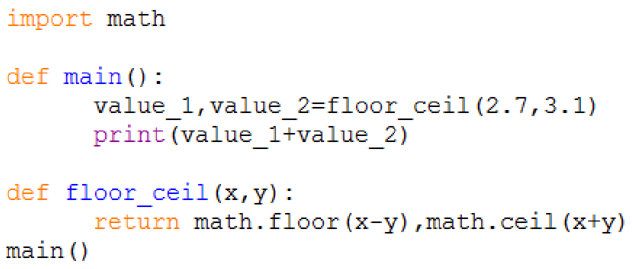
### #1



```
def main():
	if valid('amber'):
		print('yes')
	else:
		print('no')

def valid(string_in):
	is_valid=False
	if string_in[0]=='A':
		is_valid=True
	return is_valid
main()
```

### #2

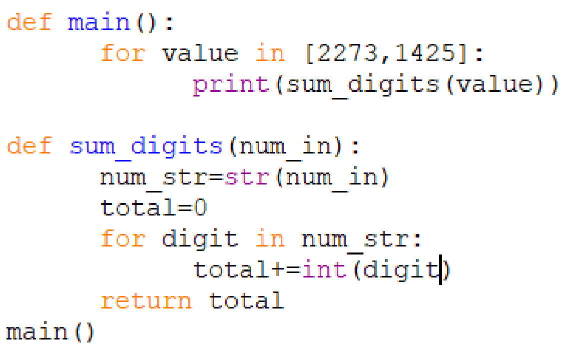


```
import math

def main():
	value_1,value_2=floor_ceil(2.7,3.1)
	print(value_1+value_2)

def floor_ceil(x,y):
	return math.floor(x-y),math.ceil(x+y)
main()
```

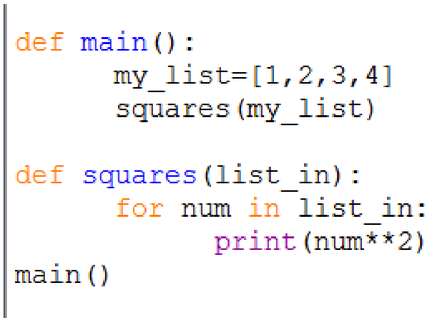
### #3



```
def main():
	for value in [2273,1425]:
		print(sum_digits(value))

def sum_digits(num_in):
	num_str=str(num_in)
	total=0
	for digit in num_str:
		total+=int(digit)
	return total
main()
```

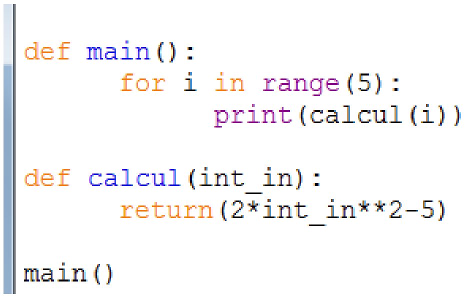
## #4



```
def main():
    my_list=[1,2,3,4]
    squares(my_list)

def squares(list_in):
    for num in list_in:
        print(num**2)
main()
```

## #5

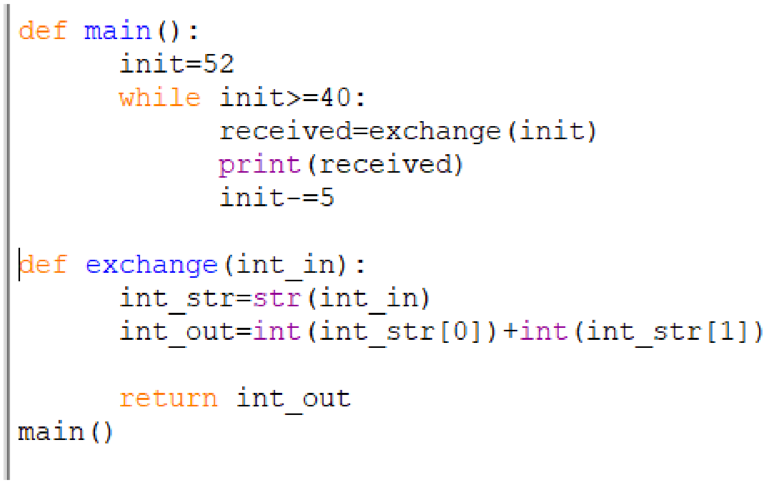


```
def main():
    for i in range(5):
        print(calcul(i))

def calcul(int_in):
    return(2*int_in**2-5)

main()
```

## #6



```
def main():
    init=52
    while init>=40:
        received=exchange(init)
        print(received)
        init-=5

def exchange(int_in):
    int_str=str(int_in)
    int_out=int(int_str[0])+int(int_str[1])

    return int_out
main()
```

## PART II. PROGRAMMING

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

**PROGRAM #1: Calories from Fat and Carbohydrates – must be submitted to CANVAS as lab8.py. Start the program with: # your name.**

A nutritionist who works for a fitness club helps members by evaluating their diets. As part of her evaluation, she asks members for the number of fat grams and carbohydrate grams that they consumed in a day. Then, she calculates the number of calories that result from the fat, using the following formula:

calories from fat = fat grams × 9

Next, she calculates the number of calories that result from the carbohydrates, using the following formula:

calories from carbs = carb grams × 4

The nutritionist asks you to write a program that will make these calculations and to display the individual calories and the total of calories.

STEP 1: Write following functions:

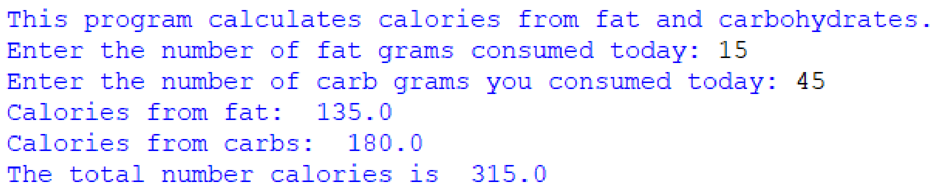
A value returning function `cal_fat` with an appropriate parameter to calculate and to return calories from fat.

A value returning function `cal_carbs` with an appropriate parameter to calculate and to return calories from carbs.

STEP 2:

Write a program which uses the written functions and prompts a user to enter fat grams and the carbohydrate grams and calculates and displays the total calories from fat and carbohydrates.

**Example of a possible output:**



```
This program calculates calories from fat and carbohydrates.
Enter the number of fat grams consumed today: 15
Enter the number of carb grams you consumed today: 45
Calories from fat:  135.0
Calories from carbs:  180.0
The total number calories is  315.0
```

**PROGRAM #2: RANDOM NUMBERS**

STEP 1.

Write a value returning function `sequence` with a parameter representing a number of integers in a series. The function:

- Creates and displays random integers from 0 to 10 (you can use or not a list);
- Returns the average value of these integers;
- Returns a number of zeros in the sequence.

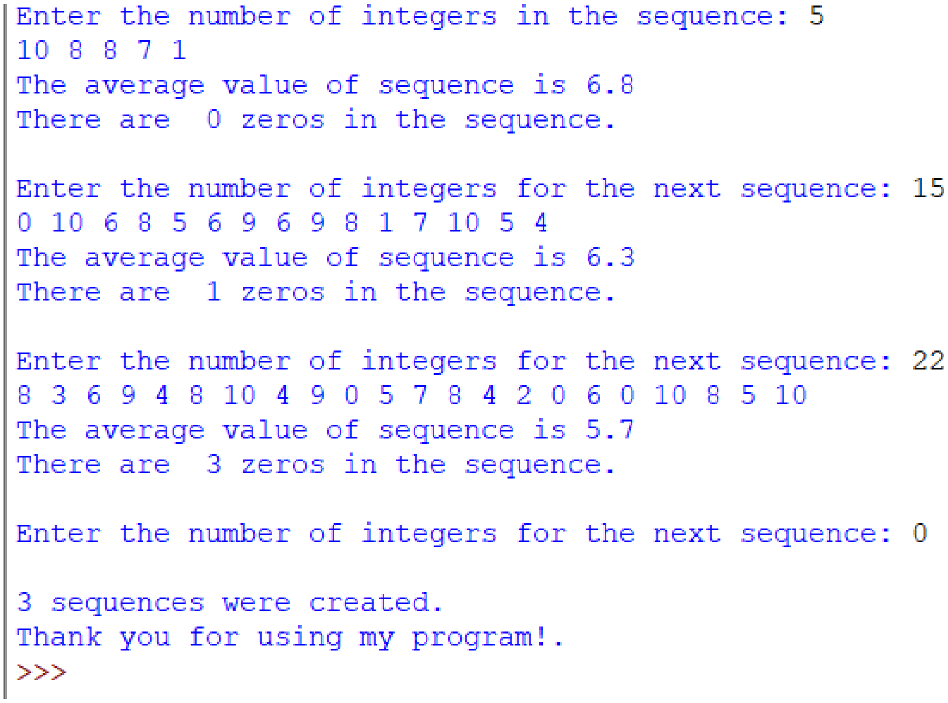
STEP 2.

Write a program which is using the created function. A user is prompt to enter a number of integers in the sequence and the program displays for the created sequence the average value and the number of zeroes in the sequence.

The program creates sequences repeatedly, by entering 0 or a negative number for number of integers in a sequence the program ends.

At the end the program displays the number of created sequences.

**See a possible output (in this version lists were not used):**



```
Enter the number of integers in the sequence: 5
10 8 8 7 1
The average value of sequence is 6.8
There are  0 zeros in the sequence.

Enter the number of integers for the next sequence: 15
0 10 6 8 5 6 9 6 9 8 1 7 10 5 4
The average value of sequence is 6.3
There are  1 zeros in the sequence.

Enter the number of integers for the next sequence: 22
8 3 6 9 4 8 10 4 9 0 5 7 8 4 2 0 6 0 10 8 5 10
The average value of sequence is 5.7
There are  3 zeros in the sequence.

Enter the number of integers for the next sequence: 0

3 sequences were created.
Thank you for using my program!.
>>>
```