Istanbul Medipol University
School of Engineering and
Natural Sciences
Computer Engineering

Fall 2019 – Assignment #8
Introduction to Programming
Assignment Date: 18.12.2019
Due Date: 30.12.2019

Programming Assignment #8

Put all of you code into the same PYHTON file according to file name conventions given in http://www.akyokus.com/fall2019/ip/assignments.html. Programming assignment solutions should be submitted in the following format:

File Name: COE-64160099-KAYA-A1.py

```
##########################################################################################
#  Name:         Ali Cokcalışır
#  Student ID:   6321211
#  Department:   Computer Engineering
#
#  Assignment ID: A1
##########################################################################################

##########################################################################################
#  QUESTION I
#  Description:
#  Body mass index (BMI) is a measure of health #  based on weight. It can be calculated by
#  taking your weight in kilograms and # dividing it by the square of your height in meters.
#  Write a program that prompts # the user to enter a weight in pounds and height in inches
#  and displays the BMI. Note that one pound is 0.45359237 kilograms and one inch is
#  0.0254 meters.
#
#  Sources:
#  Give references for the sources that you used in your program if there are any
#
##########################################################################################

print("\n")
print("SOLUTION OF QUESTION I:")
print("********************************************************************************")
# Prompt the user to enter weight in pounds
weight = eval(input("Enter weight in pounds: "))

# Prompt the user to enter height in inches
height = eval(input("Enter height in inches: "))

KILOGRAMS_PER_POUND = 0.45359237  # Constant
METERS_PER_INCH = 0.0254  # Constant

# Compute BMI
weightInKilograms = weight * KILOGRAMS_PER_POUND
heightInMeters = height * METERS_PER_INCH
bmi = weightInKilograms / (heightInMeters * heightInMeters)

# Display result
print("BMI is", format(bmi, ".2f"))
if bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal")
elif bmi < 30:
    print("Overweight")
else:
    print("Obese")
```

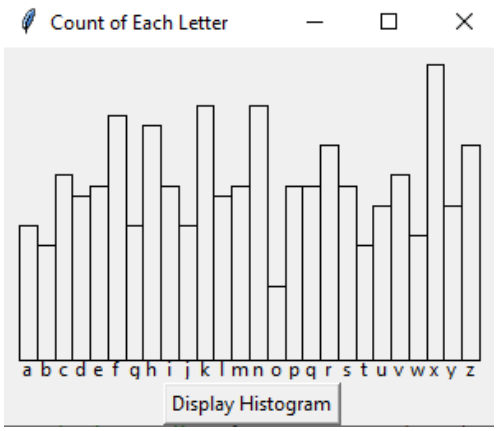
```
##########################################################################################
#  QUESTION II
#  Description:
#  You can use the math functions to solve many computational problems. Given the three
#  vertices of a triangle, for example, you can compute the angles by using
#  math formulas. The following program asks user to enter the coordinates of a triangle,
# then it computes its angles.
#
#  Sources:
#  Give references for the sources that you used in your program if there are any
#
##########################################################################################

print("\n")
print("SOLUTION OF QUESTION II:")
print("*********************************************************************************")
import math

x1, y1, x2, y2, x3, y3 = eval(input("Enter six coordinates of three points \
separated by commas like x1, y1, x2, y2, x3, y3: "))

a = math.sqrt((x2 - x3) * (x2 - x3) + (y2 - y3) * (y2 - y3))
b = math.sqrt((x1 - x3) * (x1 - x3) + (y1 - y3) * (y1 - y3))
c = math.sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2))

A = math.degrees(math.acos((a * a - b * b - c * c) / (-2 * b * c)))
B = math.degrees(math.acos((b * b - a * a - c * c) / (-2 * a * c)))
C = math.degrees(math.acos((c * c - b * b - a * a) / (-2 * a * b)))

print("The three angles are ", round(A * 100) / 100.0,
      round(B * 100) / 100.0, round(C * 100) / 100.0)


##########################################################################################
#  QUESTION III
#  Description:
#  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
#  ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
#  ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
#  reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
#  Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
#  anim id est laborum
#
#  Sources:
#  Give references for the sources that you used in your program if there are any
#
##########################################################################################
print("\n")
print("SOLUTION OF QUESTION III:")
print("*********************************************************************************")
```

Submit your programming assignment through the Google classroom IntProg-Fall2019 (https://classroom.google.com/c/NDMyNjgzODE5MjNa).

NOTE: YOU ARE ONLY ALLOWED TO SEND YOUR HOMEWORK IN '.py' OR '.zip' FILE. DO NOT FORGET TO FOLLOW THE NAMING FORMAT. DO NOT COPY CODES FROM YOUR FRIENDS. IF IT OCCURS, THEN YOU AND YOUR FRIEND WILL GET 0 (ZERO) BECAUSE OF THE PLAGIARISM, TRY YOUR BEST.

QUESTION I (20p) :

Write a program that generates 1,000 lowercase letters randomly, counts the occurrence of each letter, and displays a histogram for the occurrences, as shown in the Figure.
NOTE = You are not allowed to use RandomCharacter library.



QUESTION II (30p) :

Write the following function that sorts and merges two lists into a new sorted list:

def merge(list1, list2):

Enter list1: 10,5,6,7
Enter list2: 12,8,5,0,1
The merged list is: 0,1,5,5,6,7,8,10,12

QUESTION III (30p) :

Write a Python program that read exactly 8 numbers. Use Bubble Sort algortihm to sort it. While sorting, display each iteration.

Numbers = [45,62,39,71,77,13,44,55]

```
[45, 62, 39, 71, 77, 13, 44, 55]
[45, 62, 39, 71, 77, 13, 44, 55]
[45, 39, 62, 71, 77, 13, 44, 55]
[45, 39, 62, 71, 77, 13, 44, 55]
[45, 39, 62, 71, 77, 13, 44, 55]
[45, 39, 62, 71, 13, 77, 44, 55]
[45, 39, 62, 71, 13, 44, 77, 55]
[45, 39, 62, 71, 13, 44, 55, 77]
[39, 45, 62, 71, 13, 44, 55, 77]
[39, 45, 62, 71, 13, 44, 55, 77]
[39, 45, 62, 71, 13, 44, 55, 77]
[39, 45, 62, 13, 71, 44, 55, 77]
[39, 45, 62, 13, 44, 71, 55, 77]
[39, 45, 62, 13, 44, 55, 71, 77]
[39, 45, 62, 13, 44, 55, 71, 77]
[39, 45, 62, 13, 44, 55, 71, 77]
[39, 45, 13, 62, 44, 55, 71, 77]
[39, 45, 13, 44, 62, 55, 71, 77]
[39, 45, 13, 44, 55, 62, 71, 77]
[39, 45, 13, 44, 55, 62, 71, 77]
[39, 13, 45, 44, 55, 62, 71, 77]
[39, 13, 44, 45, 55, 62, 71, 77]
[39, 13, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
[13, 39, 44, 45, 55, 62, 71, 77]
```

QUESTION IV (50p):

In this question, you are going to write methods that operate on matrices. The program reads values of matrices A, B, and C stored in a file called inputs.txt. This file should be placed under current directory where you have the program. The first line before each matrix contains the number of rows and the number of columns as shown below.

<Input>
4 4
55 55 55 56
66 66 66 67
77 77 77 78
88 88 88 89
4 4

1 2 3 4
2 2 2 2
3 3 3 3
4 4 4 4
4 4
11 12 13 14
22 24 24 25
33 34 35 36
44 45 46 47
<End Input>

As a first step, the program reads data for matrices A, B and C from inputs.txt file and write them into console. A forth matrix D is generated randomly and printed. A partial program code is given below.

<Code>

```
import random
import sys


def readMatrix(numberOfRows , numberOfColumns, file):
    matrix = [] # Create an empty list

    for row in range(numberOfRows):
        matrix.append([]) # Add an empty new row
        line = file.readline()
        rowdata = [int(x) for x in line.split(' ')]
        for column in range(numberOfColumns):

            matrix[row].append(rowdata[column])
    return matrix

def printMatrix(matrix):
    for row in range(len(matrix)):
        for column in range(len(matrix[row])):
            print(format(matrix[row][column],"5d"), end = " ")
        print() # Print a new line

def fillMatrixRandomly(numberOfRows,numberOfColumns ):

    matrix = [] # Create an empty list

    for row in range(numberOfRows):
        matrix.append([]) # Add an empty new row
        for column in range(numberOfColumns):
            matrix[row].append(random.randint(0, 99))
    return matrix

def generateZeroMatrix(numberOfRows,numberOfColumns):
    matrix = [ [ 0 for i in range(numberOfRows) ] for j in range(numberOfColumns) ]
    return matrix

def addMatrix(A,B):
    C = generateZeroMatrix (len(A),len(A[0]))
    for row in range(len(A)):
        for column in range(len(A[row])):
            C[row][column] = A[row][column] + B[row][column]
    return C

    # Redirect standard output device (console) to output.txt file
    # print statements will write into output.txt file
    sys.stdout = open('output.txt', 'w')


    print("\nReading data from inputs.txt file in current directory\n")
    f = open("inputs.txt","r")
```

```
    # Read Matrix A
    line = f.readline()
    numberOfRows , numberOfColumns = [int (x) for x in line.split(' ')]
    A = readMatrix(numberOfRows , numberOfColumns, f)
    print(" **** Matrix A **** ")
    printMatrix(A)

    # Read Matrix B
    line = f.readline()
    numberOfRows, numberOfColumns = [int(x) for x in line.split(' ')]
    B = readMatrix(numberOfRows, numberOfColumns, f)
    print(" **** Matrix B **** ")
    printMatrix(B)

    # Read Matrix C
    line = f.readline()
    numberOfRows, numberOfColumns = [int(x) for x in line.split(' ')]
    C = readMatrix(numberOfRows, numberOfColumns, f)
    print(" **** Matrix C **** ")
    printMatrix(C)

    # Generate 4x4 matrix from random numbers.
    D = fillMatrixRandomly(numberOfRows, numberOfColumns)
    print(" **** Matrix D **** ")
    printMatrix(D)

    # Compute S = (A+B) * Transpose(C) + D - A
    print("\n *** Computing S = (A+B) * Transpose(C) + D) - A *** \n")

    #  T1 = A + B
    T1 = addMatrix(A,B)
    print(" **** MatriX T1 = (A+B) ****")
    printMatrix(T1)


main()
```

<End code>

You are going to write the code for the following methods and the missing code in main().

```
def multiplyMatrix(A, B)
def transpose(A)
def maxOfElements(A)
def subtractMatrix(A, B)
```

The program will calculate S = (A+B) * Transpose(C) + D) – A and find the maximum element in S. Complete the code given above so that it will produce an output as follows:

<Output>

```
Reading data from inputs.txt file in current directory

 **** Matrix A ****
   55    55    55    56
   66    20    12    67
   77    15    25    78
   88    12    13    89
 **** Matrix B ****
    1     2     3     4
    2     2     2     2
    3     3     3     3
    4     4     4     4
 **** Matrix C ****
   50    12    75    14
```

```
  55    24    24    25
  33    34    35    36
  44    45    46    47
 **** Matrix D ****
  19    46    14    61
  54    81    91    34
  46    60    95    40
  52    74    95    31

 *** Computing S = (A+B) * Transpose(C) + D) - A ***

 **** MatriX T1 = (A+B) ****
  56    57    58    60
  68    22    14    69
  80    18    28    81
  92    16    17    93
 **** Matrix T2 = Transpose(C) ****
  50    55    33    44
  12    24    34    45
  75    24    35    46
  14    25    36    47
 **** Matrix T3 =(A+B) * transpose(C) ****
 8674  7340  7976 10517
 5680  6329  5966  7869
 7450  7529  7148  9425
 7369  8177  7523  9921
**** Matrix T4 =(A+B) * transpose(C)+ D ****
 8693  7386  7990 10578
 5734  6410  6057  7903
 7496  7589  7243  9465
 7421  8251  7618  9952
**** Matrix S =(A+B) * transpose(C) + D - A ****
 8638  7331  7935 10522
 5668  6390  6045  7836
 7419  7574  7218  9387
 7333  8239  7605  9863

Maximum Element in S = 10522

<End Output>
```

Submit your program code and the output of your program which is stored in *output.txt* file in a ZIP or RAR file.

QUESTION V (30p):

Design a class named **Triangle** that extends the **GeometricObject** class. The **Triangle** class contains:

■ Three float data fields named **side1**, **side2**, and **side3** to denote the three sides of the triangle.
■ A constructor that creates a triangle with the specified **side1**, **side2**, and **side3** with default values **1.0**.
■ The accessor methods for all three data fields.
■ The class should throw a RuntimeError exception if the three given sides cannot form a triangle (If one of sides is o or less than 0.)
■ A method named **getArea()** that returns the area of this triangle.
■ A method named **getPerimeter()** that returns the perimeter of this triangle.
■ A method named **__str__()** that returns a string description for the triangle.

The **__str__()** method is implemented as follows:

return "Triangle: side1 = " + str(side1) + **" side2 = "** +
str(side2) + **" side3 = "** + str(side3)

Implement the **Triangle** class. Write a test program that prompts the user to enter the three sides of the triangle, a color, and **1** or **0** to indicate whether the triangle is filled. The program should create a **Triangle** object with these sides and set the color and filled properties using the input. The program should display the triangle's area, perimeter, color, and **True** or **False** to indicate whether the triangle is filled or not.

QUESTION VI (40p):

Write a program that reads in a Python source code file and counts the occurrence of each keyword and each identifier (variables, class and method names) in the file using **two dictionaries:** one for keywords and one for identifiers**.** Your program should prompt the user to enter the Python source code filename.

Keywords in Python programming language:

False	class	finally	is	return	None	continue	for	lambda	try			
True	def	from	nonlocal	while	and		del	global	not	with	as	elif
if	or	yield	assert		else	import	pass	break	except	in		raise

Note:

Submit all of your solutions within a ZIP or RAR file.