CIS3389 (Programming for Data Processing) Spring 2021
Assignment #1

Due date: 02/20/2021; 11:30 pm

Instructions: Write each of the following 4 programs in Python using IDLE. Save each program by its corresponding question number followed by team members' first names. For example, for Q1, you will save the program in a file named "Q1_xxx.py" where xxx is the first names of team members. This assignment to be completed in a team. Teams are already assigned for the course.

Deliverables: 4 Python files: Q1_xxx.py
                                    Q2_xxx.py
                                    Q3_xxx.py
                                    Q4_xxx.py

Method of Submission: Upload all the deliverables in Canvas>Assignments>Assignment #1 by the due date and time. Only one submission per team (one member from each team will upload the files).

Important:

- Late submissions will not be accepted.
- Follow the output formatting asked in the questions. If format is not done correctly, points will be deducted.

Total points: 100

1. (**20 points**)

Write a python program-

   a. to calculate and print the squares of the integers between 5 and 30 (both included) that are divisible by 5 except 10 and 15 (write the program in a way so that it should not print and calculate the squares of 10 and 15 but should print and calculate square for all other integers that are divisible by 5 and between 5 and 30).
   b. Calculate the sum of all the squares you found as part of (a).

   Format your output as follows:

Number	Square
5	25
:	:
:	:
Sum of squares: ………….	

2. (**30 points**)

The following string is given:

Sam works in a company abc in New York. He joined the company in 2019. Before joining ABC, he used to work for a small firm in Arizona. He worked there from 2015 to 2018. Before moving to Arizona Sam used to live in South Dakota and he has been living there since 2000's.

Now perform the following tasks:

a. Write a python program to create and print a python list with only the year values mentioned in the above string. For example, 2019 , 2015 …. are some examples of year values in the above string.
b. Determine and print how many times the substring 'abc' appear in the above string irrespective of upper or lower cases. Also calculate the maximum and minimum of the year values from the list you created in a.
c. Replace all 'h' with '_' in the above string and print the new string. (Note: if you print the new string all in lower case or upper case that is fine)

Now, print all the outputs from a. to c. in a text file called "Q2Output.txt".

Your output in the "Q2Output.txt" file should look like (make sure your answer format is exact same as below) :

a. The list of years: …………
b. Count: …… , Maximum: ……., Minimum:………
c. The new string: ……………….

3. (**25 points**)

You have been given the following dictionary of state names:

dic1 = {'L1': ['NY', 'CT', 'NH', 'MA'], 'L2': ['TX', 'NM'], 'L3': ['CA', 'WA', 'AZ'], 'L4': ['ND', 'SD','WY', 'ID'], 'L5':['UT'],'L6':['MN','WI','KY']}

Write a Python program that prints all the keys along with the count of states for each key from the above dictionary to a text file Q3Output.txt.

Your output in the text file should look like:

L1: Number of states – 4

L2: Number of states – 2

L3: Number of states - 3

:

4. (**25 points**)

Write a Python program to determine if Lina will be within her budget, over her budget or exactly matches her budget next month.
The inputs will be:

Budget: Ask Lina's next month's budget in $.

Expenses: For the next month's expenses, the program should ask two things separately as user inputs -
a. next month's house rent in $ and

b. all other expenses for next month in $

After all the inputs are entered: Once Lina enters both the expenses, the program should calculate Lina's total expenses for next month. Then, the program will check whether Lina's total expenses for next month would be less than the next month's budget, equals to the next month's budget or greater than the next month's budget.
Write the program in a way so that program keeps asking for inputs and shows the corresponding outputs if user says "y" [y=yes] and quits if user says "n" [n=no]

Your output should look like:

Lina's budget for next month in $:

Lina's total expenses for next month in $:

Status: over/under/matches budget

Continue? y/n