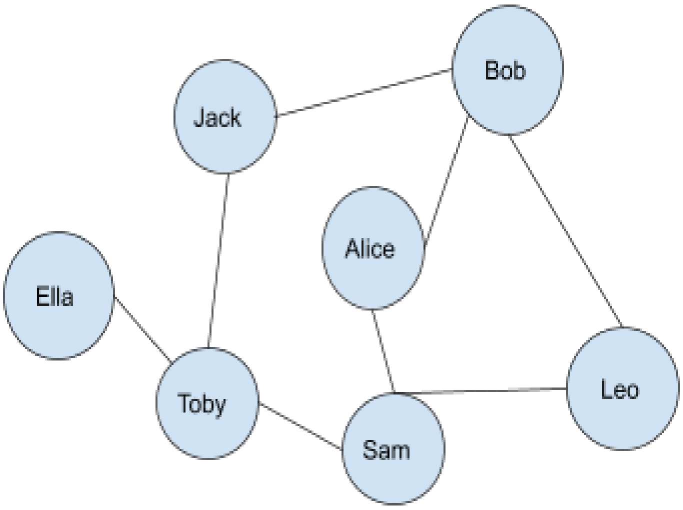
Due date: 08.12.2019, 23:55

CENG 113 Programming Basics
HOMEWORK #3
27.11.2019

A social network consists of individuals and the relationships between them. Let us consider a simple network of friends which includes seven users: Alice, Bob, Jack, Sam, Leo, Ella, and Toby, and say that Alice is friends with Bob and Sam (so, Bob and Sam are automatically friends of Alice), Bob is a friend of Jack (so, Jack is a friend of Bob), etc. Graphically, we can visualize this network as below:



Each user in the network is represented with a circle and a friendship relationship between two users is displayed as a line. Note that user names are unique identifiers. A user's connections can be expanded with other users by utilizing the associations for personal and business reasons, like in Facebook or LinkedIn.

In this assignment, you will write a program that implements such a social network. Your program will enable the following functions:

- add a user: When a user is being added, this function checks whether a user with that name already exists or not, and performs the operation depending on it. **For each function, you have to check and handle all the cases.**
- delete a user: A user is deleted.
- add a friend: A friend is added to friends of the user (and vice versa).
- remove a friend: A friend is removed from friends of the user (and vice versa).
- offer a friend to a user: There are several measures to offer a friend to a user from his/her all nonfriends. For this assignment, the following two measures and an average of them will be used for each non-existent connection in the network:

- ○ The first measure is the number of mutual users that both the two users are connected to. In the example network, Sam and Bob have two mutual users, Sam and Jack have one mutual friend, etc.
- ○ The second measure divides the first measure by the total number of neighbours of the two users. If we consider Sam and Bob, they have four neighbours in total. Hence, the second measure for this candidate connection is 0.5.

For each measure, the values are normalized to scale them to [0,1] with the formula **x_scaled = (x – min) / (max – min)**. After normalization, the average value of each non-existent connection is found. Next, the potential friends of the entered user should be selected, and sorted by the average measure, displayed on the console, and the offered friend is returned. The offered friend to the user Sam is Bob, as can be seen below:

```
Potential friends of the user Sam:
'Bob', 'Sam', 0.75
'Sam', 'Ella', 0.417
'Jack', 'Sam', 0.375
```

The important part of this assignment is to decide **which data structures** are suitable to store the social network and the calculated three measures. You must use the data structures that were covered at the lectures and laboratories. You are not allowed to use classes or recursions (any topic that was not covered) to be fair to everyone.

There will be 3 python files inside the zip file you will send. The first file named as "**ceng113_hw3_studentID.py**" will contain just the function definitions, the second file named as "**ceng113_hw3_studentID_test.py**" will build up the example social network by calling the functions of the first file, and offer and print a friend to the user Sam, and the third file named as "**ceng113_hw3_studentID_menu.py**" will have a command-line menu with the five options (an option per function). To call the defined functions in a python file from another python file, you should add "**import ceng113_hw3_studentID as hw3"** (without quotes) at the beginning of the caller file, and call the functions with "hw3." (without quotes). For example, to call the function add_user, write hw3.add_user(parameters).

Submission Rules

- You have to submit your solution to CMS until due date.
- Your homework should be named as "**ceng113_hw3_studentID.zip**".
- Write your student ID as a comment at the beginning of your code.
- Cheating, including teamwork, will not be tolerated.