In this workshop, you'll write a program that works with data similar to social media basic user data in order to produce friend suggestions for users. Your workshop leader will guide you through the process. Please ***do not*** *work ahead.*

0.1 Background

Assume you are given the file `user_data.csv` that contains user screen names, first names, last names and friends of the user listed by screen name.

For example the file may contain the following:

```
screen_name,first_name,last_name,friends
CountD,Count,Dracula,Renfield
ProfHelsing,Abraham,Van Helsing,DrJ
JHarker,Jonathan,Harker,MinaM
MinaM,Wihelmina,Murray,"JHarker, LWestenra"
LWestenra,Lucy,Westenra,"MinaM, DrJ, AHolmwood"
DrJ,John,Seward,"Renfield, LWestenra"
AHolmwood,Arthur,Holmwood,LWestenra
QMorris,Quincey,Morris,LWestenra
Renfield,R.,Renfield,"CountD, DrJ"
```

The first row specifies the order of the information.

The values are separated by commas. The friends list is surrounded by quotations marks.

Our goal is to make friend suggestions for the users in the file. A user's friends suggestions are based on the friends of the user's current friends.

1 Reading data

1. Write a function, called `get_file_contents` that takes a filename and returns a list of strings where each string contains one line of the file. If the file does not exist the function should print a useful message to the user about which file was not found and return `None`.

2. Write a function called `get_items` that takes a string and returns a list of the values in the string. The sublist should contain each of the strings in the given string that were separated by commas, all values should be free from leading and trailing spaces. For example:

   ```
   get_items('screen_name,first_name,last_name,friends\n')
   ```

   would return:

   ```
   ['screen_name', 'first_name', 'last_name', 'friends']
   ```

3. Write a function called `regroup_quoted_list` that groups items in a list into sublists if the items are string containing a quote or if the item appears between two items containing a starting or an ending quote. You may assume quotes are not used in the data except to delimit the friends list. For example:

   ```
   regroup_quoted_list(['MinaM', 'Wihelmina', 'Murray', '"JHarker', ' LWestenra"'])
   ```

   would return:

   ```
   ['MinaM', 'Wihelmina', 'Murray', ['JHarker', 'LWestenra']]
   ```

4. Write a function called `get_values` that takes a list of strings and divides each string into its own list based on commas and quotes. The function should first use the function `get_items` to divide the list based on the position of the commas, and then use the function `regroup_quoted_list` to regroup the quoted items into a list of their own. For example,

   ```
   get_values(['MinaM,Wihelmina,Murray,"JHarker,LWestenra"\n',
   'LWestenra,Lucy,Westenra,"MinaM,DrJ,AHolmwood"\n'])
   ```

   would return:

   ```
   [['MinaM', 'Wihelmina', 'Murray', ['JHarker', 'LWestenra']],
   ['LWestenra', 'Lucy', 'Westenra', ['MinaM', 'DrJ', 'AHolmwood']]]
   ```

5. Write a function called `create_user_dictionary` that takes a list of strings and returns a dictionary where the keys are the user screen names and the values are themselves dictionaries that contain the user data based upon the keys that appear in line one of the 'user_data.csv' file. For instance,

```
keys = ['screen_name', 'first_name', 'last_name', 'friends']
data = [['MinaM', 'Wihelmina', 'Murray', ['JHarker',
'LWestenra']], ['LWestenra', 'Lucy', 'Westenra', ['MinaM',
'DrJ', 'AHolmwood']]]
create_user_dictionary(keys,data)
```

would return,

```
{'MinaM': {'screen_name': 'MinaM', 'first_name':
'Wihelmina', 'last_name': 'Murray', 'friends': ['JHarker',
'LWestenra']}, 'LWestenra': {'screen_name': 'LWestenra',
'first_name': 'Lucy', 'last_name': 'Westenra', 'friends':
['MinaM', 'DrJ', 'AHolmwood']}}
```

2 Searching data

6. Write a function, called `suggest_friends` that takes a user name string and a user dictionary. The function should return a list containing friend suggestions for the provided user. Friend suggestions are based on the friends of friends of the provided user. The function will **not** suggest friends of friends of friends. For example:

```
user_data = {'JHarker', 'first_name': 'Jonathan',
'last_name': 'Harker', 'friends': ['MinaM']}, 'MinaM':
{'screen_name': 'MinaM', 'first_name': 'Wihelmina',
'last_name': 'Murray', 'friends': ['JHarker',
'LWestenra']}, 'LWestenra': {'screen_name': 'LWestenra',
'first_name': 'Lucy', 'last_name': 'Westenra', 'friends':
['MinaM', 'DrJ', 'AHolmwood', 'QMorris']}}
suggest_friends('MinaM',user_data)
```

would return:

```
['DrJ', 'AHolmwood', 'QMorris']
```

7. Write a function called `friend_suggestions` that takes a dictionary of user data. The function should return a new user_data dictionary with an additional key in each user's personal info dictionary called 'friend_suggestions' that contains a list of friend suggestions. For example,

```
user_data = {'CountD': {'screen_name': 'CountD', 'first_name': 'Count',
'last_name': 'Dracula', 'friends': ['Renfield']}, 'ProfHelsing': {'screen_name':
'ProfHelsing', 'first_name': 'Abraham', 'last_name': 'Van Helsing', 'friends':
['DrJ']}, 'JHarker': {'screen_name': 'JHarker', 'first_name': 'Jonathan',
'last_name': 'Harker', 'friends': ['MinaM']}, 'MinaM': {'screen_name': 'MinaM',
'first_name': 'Wihelmina', 'last_name': 'Murray', 'friends': ['JHarker',
'LWestenra']}, 'LWestenra': {'screen_name': 'LWestenra', 'first_name': 'Lucy',
'last_name': 'Westenra', 'friends': ['MinaM', 'DrJ', 'AHolmwood']}, 'DrJ':
{'screen_name': 'DrJ', 'first_name': 'John', 'last_name': 'Seward', 'friends':
['Renfield', 'LWestenra']}, 'AHolmwood': {'screen_name': 'AHolmwood',
'first_name': 'Arthur', 'last_name': 'Holmwood', 'friends': ['LWestenra']},
'QMorris': {'screen_name': 'QMorris', 'first_name': 'Quincey', 'last_name':
'Morris', 'friends': ['LWestenra']}, 'Renfield': {'screen_name': 'Renfield',
'first_name': 'R.', 'last_name': 'Renfield', 'friends': ['CountD', 'DrJ']}}
friend_suggestions(user_data)
```

would return:

```
{'CountD': {'screen_name': 'CountD', 'first_name': 'Count', 'last_name':
'Dracula', 'friends': ['Renfield'], 'friend_suggestions': ['DrJ']},
'ProfHelsing': {'screen_name': 'ProfHelsing', 'first_name': 'Abraham',
'last_name': 'Van Helsing', 'friends': ['DrJ'], 'friend_suggestions':
['Renfield', 'LWestenra']}, 'JHarker': {'screen_name': 'JHarker', 'first_name':
'Jonathan', 'last_name': 'Harker', 'friends': ['MinaM'], 'friend_suggestions':
['LWestenra']}, 'MinaM': {'screen_name': 'MinaM', 'first_name': 'Wihelmina',
'last_name': 'Murray', 'friends': ['JHarker', 'LWestenra'],
'friend_suggestions': ['DrJ', 'AHolmwood']}, 'LWestenra': {'screen_name':
'LWestenra', 'first_name': 'Lucy', 'last_name': 'Westenra', 'friends':
['MinaM', 'DrJ', 'AHolmwood'], 'friend_suggestions': ['JHarker', 'Renfield']},
'DrJ': {'screen_name': 'DrJ', 'first_name': 'John', 'last_name': 'Seward',
'friends': ['Renfield', 'LWestenra'], 'friend_suggestions': ['CountD', 'MinaM',
'AHolmwood']}, 'AHolmwood': {'screen_name': 'AHolmwood', 'first_name':
'Arthur', 'last_name': 'Holmwood', 'friends': ['LWestenra'],
'friend_suggestions': ['MinaM', 'DrJ']}, 'QMorris': {'screen_name': 'QMorris',
'first_name': 'Quincey', 'last_name': 'Morris', 'friends': ['LWestenra'],
'friend_suggestions': ['MinaM', 'DrJ', 'AHolmwood']}, 'Renfield':
{'screen_name': 'Renfield', 'first_name': 'R.', 'last_name': 'Renfield',
'friends': ['CountD', 'DrJ'], 'friend_suggestions': ['LWestenra']}}
```

8. Write a function called `display_friend_suggestions` that takes a user data dictionary with friend suggestions and for each user prints the First and Last Name of each friend suggestion. For example, with the above user data with friend suggestions:

   ```
   display_friend_suggestions(user_data)
   ```

   would return:

   ```
   Suggested friends for Count Dracula:
   John Seward

   Suggested friends for Abraham Van Helsing:
   R. Renfield
   Lucy Westenra

   Suggested friends for Jonathan Harker:
   Lucy Westenra

   Suggested friends for Wihelmina Murray:
   John Seward
   Arthur Holmwood

   Suggested friends for Lucy Westenra:
   Jonathan Harker
   R. Renfield

   Suggested friends for John Seward:
   Count Dracula
   Wihelmina Murray
   Arthur Holmwood

   Suggested friends for Arthur Holmwood:
   Wihelmina Murray
   John Seward

   Suggested friends for Quincey Morris:
   Wihelmina Murray
   John Seward
   Arthur Holmwood

   Suggested friends for R. Renfield:
   Lucy Westenra
   ```

9. Write a `main` function that starts with the `user_data.csv` file and displays friend suggestions for each of the users in the file.