Lab 12: File Processing & Dictionaries

Labs are graded for participation rather than correctness. Keep all your lab code in your course GitHub repo to receive credit for your work. We'll be looking to see that you have at least partially completed all problems in each lab.

If you finish the lab assignment early, you may get started on the homework.

Professional skills

This is the last lab so, if you have not yet participated in collaborative coding, consider collaborating for this lab. We'll set up some breakout rooms for you to self-select to work with other students. See lab 7 for setup instructions. You can also earn professional skills points by volunteering to walk through your lab solution toward the end of class.

The Assignment

The purpose of this lab is to give you more practice with file processing and dictionaries. There is an optional extension to give you more practice with classes as well. For this lab, you're making a Facebook just for the Seven Dwarves.

Copy and paste the following text into a file named `dwarves.txt`. There are seven lines in this file, each representing one of the 7 Dwarves and all his buddies on Facebook, like this:

```
Happy Dopey Bashful Sneezy Sleepy Doc Grumpy
Dopey Bashful Sneezy Sleepy Doc Grumpy Happy
Bashful Sneezy Sleepy Doc Grumpy Happy Dopey
Sneezy Sleepy Doc Grumpy Happy Dopey Bashful
Sleepy Doc Grumpy Happy Dopey Bashful Sneezy
Doc Grumpy Happy Dopey Bashful Sneezy Sleepy
Grumpy Happy Dopey Bashful Sneezy Sleepy Doc
```

At the start, everyone is friends with everyone. In the first line, for example, we see that Happy's friends are Dopey, Bashful, Sneezy, Sleepy, Doc and Grumpy. You may assume that everything is space-separated, that each Dwarf has exactly one name, and that all friend relationships are valid.

When the program starts, read the contents of `dwarves.txt` and use it to populate the network. Use a dictionary to represent the network. Each key should be a dwarf's name. You can decide what each value should be.

Your program will ask the user which dwarf is logging in (no password required), and then print a menu with the following options:

- Print your list of friends
- Unfriend someone
- Friend someone
- Quit

When the user quits, save any changes they've made to `dwarves.txt`. Be sure to save the details of the entire network, not just the user who was interacting with your program.

We're getting practice with files today, so you must:

- Close every file that you open.
- Check for error when working with files using a try/except block.
- Make sure your program works if you run it multiple times in a row -- the file should have different data after being run, but the format should be the same, so it can still be processed.
- Handle edge cases when friending/unfriending, like the Dwarf not existing, the Dwarf not having any friends, the friend they're asking to remove not existing, etc. You can assume that all of the friend lists in the text file are valid.

Here is an example of running the program as Grumpy (user input in **blue**). First, Grumpy prints all of his friends (the other six):

```
Which of the 7 dwarves is logging in?
Grumpy
Choose from one of the options below:
P: Print your friends list
U <name>: Unfriend someone
F <name>: Friend someone
Q: Quit
p
Your friends: Dopey, Sneezy, Doc, Bashful, Sleepy, Happy
```

Then Grumpy wants to do some unfriending:

```
Choose from one of the options below:
P: Print your friends list
U <name>: Unfriend someone
F <name>: Friend someone
Q: Quit
U everyone
Everyone is not your friend.
```

```
Choose from one of the options below:
P: Print your friends list
U <name>: Unfriend someone
F <name>: Friend someone
Q: Quit
U Happy
Happy has been unfriended.
Choose from one of the options below:
P: Print your friends list
U <name>: Unfriend someone
F <name>: Friend someone
Q: Quit
p
Your friends: Sleepy, Sneezy, Bashful, Dopey, Doc
```

Now, if Grumpy quits and logs in again, his friend list is updated because it was saved in a file.

```
Which of the 7 dwarves is logging in?
Grumpy
Choose from one of the options below:
P: Print your friends list
U <name>: Unfriend someone
F <name>: Friend someone
Q: Quit
p
Your friends: Sneezy, Sleepy, Dopey, Doc, Bashful
```

Optional extension: Dwarf Facebook with classes

Make a new version of the Dwarf Facebook using classes. The functionality should be exactly the same but this time, write a class to represent a Dwarf and a class to represent DwarfFacebook. You will also need a main.py to act as the “driver” of your program. main.py should be responsible for reading and writing the text file and populating the DwarfFacebook with Dwarf objects.