

Name: Rafael Wang  
USC Student ID: 6189106477  
Email: [rafaelwa@usc.edu](mailto:rafaelwa@usc.edu)

Name: Anthony Chandra  
USC Student ID: 1245854914  
Email: [ac68801@usc.edu](mailto:ac68801@usc.edu)

Name: Xinsong Fan  
USC Student ID: 7072371273  
Email: [fanr@usc.edu](mailto:fanr@usc.edu)

Program Summary: In this phase, we expanded on the program created in Phase 1. To begin, we created a new option in the menu that allows the connection of friends, which was achieved by a new vector attribute in Person class. We also created a method to create IDs for each friend, which was used to update the print\_person method. Lastly, we have a print\_friends method that prints an individual's friends in sorted order.

C++ version: C++11

Explanation:

In terms of important structure, our enhancements to the LoadDB method use a `vector<vector<pair<string, string>>>`

that is used as a 2D vector to first hold vectors of each person's list of friends (in first and last name format). After having loaded every person into the linked list, then I use this vector to search for each individual's list of friends in the network to add to the respective myfriends vector.

To sort each individual's friends, we used selection sort on each friend's ID code.

References:

No references were used.

Assumptions:

We made the assumption that when loading in the input txt file, the friends' names would be in format like "Amy, Chen" or "Kurt, Cobain", similar to that of Part 5.

Instructions:

To run this, first ensure all files are in the same directory. Then, based on what the main program is (assuming here it's test\_network.cpp), then we run in the terminal after locating the correct directory:

```
g++ -std=c++11 test_network.cpp network.cpp person.cpp contact.cpp date.cpp misc.cpp  
./a.out
```

This will bring up the menu and the user can choose which option to select.

Some test cases and results:

One test case we tried is first loading in a text file using the new loadDB method (where some people may have a list of friends), and then using the new option 6 to connect friends, which is shown below.

```
Make friends:  
Person 1  
First Name: Julia Scarlett Elizabeth  
Last Name: Louis-Dreyfus  
Person 2  
First Name: Webs  
Last Name: Burg  
  
Louis-Dreyfus, Julia Scarlett Elizabeth  
January 13, 1961  
Phone (Home): 310-192-2011  
Email (Work): julia@wh.com  
  
Burg, Webs  
January 20, 1999  
Phone (Home): 408-456-7814  
Email (School): WB@usc.edu
```

Then, another test case we tried is using the updated saveDB method, and the output txt file would look something like below, where it also demonstrates the use of the codeName method:

```
Louis-Dreyfus, Julia Scarlett Elizabeth
January 13, 1961
Phone (Home): 310-192-2011
Email (Work): julia@wh.com
shawnzha0 (Shawn Zhao)
websburg (Webs Burg)
-----
Zhao, Shawn
August 19, 1999
Phone (Home): 408-456-2891
Email (School): sz@usc.edu
juliascarlettelizabethylouis-dreyfus (Julia Scarlett Elizabeth Louis-Dreyfus)
-----
Burg, Webs
January 20, 1999
Phone (Home): 408-456-7814
Email (School): WB@usc.edu
juliascarlettelizabethylouis-dreyfus (Julia Scarlett Elizabeth Louis-Dreyfus)
-----
```

The `print_friends` method was not used in the program menu itself, but we tested it on our own and it does indeed print each friend in the correct sorted order.