Ashly Lam, Amanda Poon
4181603610, 9451754209
ashlylam@usc.edu
ajpoon@usc.edu

The program creates a network, or doubly linked list, containing each person's personal information. This personal information includes their first and last name, birthdate, phone number, email address, and can include the type of phone number and email address. Through the menu, users can save the network database to a .txt file, load .txt files of people's information to add to the network database, add and remove individual users, and print users.

network.cpp
This file displays a menu for the user to interact with. Through the menu, users can save the network database to a .txt file, load .txt files of people's information to add to the network database, add and remove individual users, and print users.

To save the network database to a .txt file, the saveDB function first opens a file that is named according to the user input. Then, it goes through the linked list and outputs the information of each person into the file.

To load the information in the file to the network database of a doubly linked list, it first checks that the file exists. If the file exists, the function loadDB uses getline to read each line of the file and assign it to its matching variable. We used getline because there could be spaces in the name and in separating the type from the phone number and email address. We also incremented count each time a person was added to the network because count keeps track of how many people there are on the network. Each time a person is added, they are added to the linked list.

To add an individual, it first checks that the person exists in the database given the person's first name and last name. Using the first and last name, the search function goes through the linked list to check if each person in the network has a matching first and last name to the user-provided first and last name. If the person exists in the database, additional information including the birthday, email, and phone number of the person are received as input, a new person is created and the push_front function adds them to the front of the linked list.

To remove an individual, it first checks that the person exists in the database given the person's first name and last name. Using the first and last name, the search function goes through the linked list to check if each person in the network has a matching first and last name to the user-provided first and last name. If the person exists in the database, the remove function goes through the linked list to check which person has a first and last name that match the user input. The prev and next pointers in the database update the next and prev pointers they point to, and then the user input person is freed.

To print the people given the last name, getline gets the last name. Then, it goes through the linked list to see if any person has a matching last name. If no one has a matching last name, it prints “Nothing matched!”

To connect with friends, it checks that both of the user-inputted people exist in the database, adds each as friends in a vector, and then prints each person’s list of friends in alphabetical order as well as in their IDs.

The user can continue to interact with the menu until they enter an invalid input.

contact.cpp
Includes constructor for Email and Phone given their type and email address or phone number. The set_contact functions cin the type and email address or phone number. The get_contact functions return the email address or phone number and type if applicable in the formatted information. The print functions print the formatted information.

date.cpp
The Date constructor parses the argument to identify the month, day, and year. The print_date function prints the formatted date. The get_date() function returns the formatted date as a string. The get_month, get_day, and get_year functions according return the month, day, and year.

misc.cpp
Prints the banner.

person.cpp
There are three types of constructors. Default constructor, constructor with names and birthday and email and phone as arguments, and constructor with file as an argument. The set_person() function uses user input to assign the personal information to the person and the other set_person function(filename) uses the file to assign the personal information to the person. The operator== and operator!= functions check if the two people match and return the boolean value. print_person prints the people. A vector stores the person’s list of friends. Two functions print the list of friends; one prints the code and name of the friend and the other function prints the names of the friends in alphabetical order.

.h files contain the functions and attribute declarations of each class.

Instructions

g++ contact.cpp date.cpp misc.cpp network.cpp person.cpp test_network.cpp
./a.out


ashlylam@Ashlys-MacBook-Pro project1 % g++ person.cpp contact.cpp date.cpp
ashlylam@Ashlys-MacBook-Pro project1 % ./a.out
Louis-Dreyfus, Julia Scarlett Elizabeth
January 13, 1961
(Home) 310-192-2011
(Work) julia@wh.com
Louis-Dreyfus, Julia Scarlett Elizabeth
January 13, 1961
(Home) 310-192-2011
(Work) julia@wh.com
Is c1 equal to c2: 1
Is c1 not equal to c2: 0
ashlylam@Ashlys-MacBook-Pro project1 %