

CSCI 2270 – Data Structures and Algorithms
Instructor: Hoenigman
Lab 4
Due: Monday, July 14 by 7pm

Communication between towers

In the Lord of the Rings trilogy, there is a scene where the first beacon is lit in the towers of Minas Tirith. The second beacon then sees the fire, and knows to light its fire to send a signal to the third beacon, and so forth. This was a means of communicating in the days before telegraphs were invented as it was much faster than sending a human rider to deliver a message. Communication towers were equipped with signaling mechanisms, such as mirrors, that could spell out messages using the positions of the mirrors.

Today, there are several examples of communication networks that are conceptually similar but much more technically advanced that route messages through multiple hubs between the sender and the receiver. For example, when you type a URL into a web browser, a request is sent through a network of service providers to the destination, and then packets of information are sent back to your machine. If I type www.google.com from my home in Boulder, my request follows this path:

```
1  192.168.2.1 (192.168.2.1)
2  c-24-9-60-1.hsd1.co.comcast.net (24.9.60.1)
3  te-9-7-ur02.boulder.co.denver.comcast.net
4  xe-13-3-1-0-ar01.aurora.co.denver.comcast.net
5  he-3-10-0-0-cr01.denver.co.ibone.comcast.net
   (68.86.92.25)
te-1-1-0-4-cr01.chicago.il.ibone.comcast.net (68.86.95.205)
6  xe-2-0-0-0-pe01.910fifteenth.co.ibone.comcast.net
   (68.86.82.2)
7  as15169-1-c.910fifteenth.co.ibone.comcast.net
   (23.30.206.106)
8  72.14.234.57 (72.14.234.57)
9  209.85.251.111 (209.85.251.111)
10 den03s06-in-f16.1e100.net (74.125.225.208)
```

Each IP address is a hop in the network for my request, which is received at each service provider the service provider and then forwarded to the next service provider in the network, depending on the final destination of the message.

(Note: I got this path by typing `tracert www.google.com` in a terminal window. From campus, you will see a different path.)

Build your own communications network

In this assignment, you're going to simulate a communications network using a linked list. Each node in your linked list will represent a city and you need to be able to send a message between nodes from one side of the country to the other.

Include the following cities in your network:

Los Angeles
Phoenix
Denver
Dallas
St. Louis
Chicago
Atlanta
Washington, D.C.
New York
Boston

Implement each city as either a struct or a class, either one will be accepted for full credit on this lab. You could also create a `LinkedList` class and a `City` class and set up your `LinkedList` to include cities.

Your cities need to be identified by name, and have a forward path connecting it to the next city in the network, and a place to store the message being sent. (You can assume the message is a string.)

There is a file on Moodle called *messageIn.txt* that contains one line of text with words separated by spaces. Your program needs to transmit each word one at a time from Los Angeles to Boston, going through each city in the linked list. When a word is received in Boston, it should be printed to the terminal. Perform garbage collection and end the program.

Transmitting the message between cities needs to be done in a separate function called *transmitMsg*. If you are using classes, *transmitMsg* should be part of your class. If you are using structs, it will be a function. When you read a word from the file, call the *transmitMsg* function and assign the word to the first city. You then need to call the *transmitMsg* function each time you send the word to the next city. Keep in mind that transmitting a message means that only one city can hold the message/word at one time. As soon as the message is transmitted to the next city, the message should be removed from the sender city.

Implementation details

You need to create a class and a header file for your city and include the header file in your program. To submit your work, zip all three files together and submit the zip file. All files should include a comment block at the top of the file with your name, instructor's name, and lab number.