CSC230 Lab 6

Goal: This lab includes function and linked list, fstream, and eof().

Please try your best to finish the lab in class, and submit it to CANVAS.

In this lab, please write a **Lab6.cpp** file. There is a main function in this file. Please define one or two more function in the lab. The new function defined by you should be append() and search().

In Lab4, we read contents from a file and save the contents to an array. In this lab, we will repeat the functionality implemented by Lab4, but with a **singly linked list**, instead of array. Please note that you MUST use singly linked list to implement this lab. If the code is implemented with array or vector, or any data structure other than singly linked list, the grade will be 0!

In the lecture of linked list, we introduced a program on slides page 9 and 20. Please carefully read the code, and fully understand the code.

Just like Lab4, Lab6.cpp reads contents from an input file, which is "sample.txt" in today's lab. Each row of the input file has a leading char, followed by a **string** of SSN, and first name and last name. Whenever Lab6.cpp reads one row from the file, it stores SSN and the corresponding name (including both first name and last name) to a singly linked list. After the whole input file is processed, the program prompts the user to type a SSN, then it will search the SSN in the singly linked list. If there is a match, the program prints out the index value of the node. Suppose the first node has index value 0. For example,

```
jli$ ./a.out sample.txt
Input a SSN:
766434955
Found at location 4

jli$ ./a.out sample.txt
Input a SSN:
038249140
Found at location 8
```

In this lab, you can use fstream library to read file. An example code is listed as follows:

```
#include <iostream>
#include <fstream>
using namespace std;

int main(int argc, char* argv[]){
  int x, y, z;
  fstream input(argv[1]);
```

```
  while(!input.eof()){
    input >> x >> y >> z;
  }
  input.close();
}
```

If you do not remember how to read the contents from a file, refer to your lab4 implementation.

Hints & **Requirements**

- The append() function MUST insert the node to the **end** of the list, NOT the beginning of the list or any middle position of the list.
- The Node defined in the slides has only one user variable, *val*. You can define multiple variables inside one node. For example, you can define *string SSN;* and *string name;* inside the node.
- Do NOT define any array in this lab. You do not need array in this lab.
- search() function is optional. You can define a separate search() function to search a given SSN, or write a loop inside the main() function doing the searching job.

Wrap up

When you're done, zip three files to Lab6.zip

zip Lab6.zip *

Submit Lab6.zip to Canvas.

Make sure you logout before you leave!

If you cannot finish lab in class, please save all your files. Next time you login the computer in the lab, you can continue work on your files. Please save them before you logout. If you work in a Linux lab, please save the file to your machine. However, if you are working in the Mac lab, please save the file to a CLOUD. The Mac machine will erase everything you saved once you logout.