CS 162 LAB #10 – Recursion & List

In order to get credit for the lab, you need to be checked off by the end of lab. Since this is the very last lab of the course, you are not able to make up points afterwards. For extenuating circumstances, contact your lab TAs and the instructor.

This lab is worth 15 points total. Here's the breakdown:

- 5 points: Worksheet
- 5 points: Solve a problem using recursion
- 5 points: Implement question on a singly linked list

(5 pts) Part 1: Worksheet

This session will be led by your lab TAs. Please follow their instructions, participate, and complete worksheet 10:

https://classes.engr.oregonstate.edu/eecs/winter2024/cs162-001/labs/WS10.docx (pdf version)

To start, download the zip file from:
wget https://classes.engr.oregonstate.edu/eecs/winter2024/cs162-001/labs/lab10.zip

To extract it:
```
unzip lab10.zip
```

(5 pts) Recursion

Problem Statement: Given a positive integer N, count all possible distinct binary strings of length N such that there are no consecutive 0's.

Example 1:

```
Input: N = 2

Output: 3

Explanation: The 3 strings are 01 10 11
```

Example 2:

```
Input: N = 3

Output: 5

Explanation: The 5 strings are 010 011 101 110 111
```

Complete the no_zero_strings() function in recursion.cpp. It accepts N as a parameter and returns the number of distinct strings of length N without consecutive 0's. Note that it does not have to print out all of the allowed strings; it only has to compute and return the number of the strings.

It's true that there's an efficient non-recursive solution to this problem, **but you must implement a recursive solution.**

Tips:

Let S_N be the number of distinct strings of length N without consecutive 0's.

1. In order to compute S_N, what needs to be true for S_{N-1}?
2. What is the relationship between S_k and "number of strings ending with 1 in S_{k-1}?
3. What is the relationship between S_k and "number of strings ending with 0 in S_{k-1}?

(5 pts) Singly Linked List

In this part, you will implement a linked list class using pointers and object-oriented programming Although the C++ STL (Standard Template Library) offers a linked list implementation, you must implement this program "from scratch" and cannot simply utilize the existing STL offerings (i.e., the use of <list> or <forward_list> are not allowed!).

Your linked list will be designed to contain signed integers of type **int**.

Use node.h, linked_list.h, linked_list.cpp and test_linked_list.cpp from the zip file.

Required Class:

You must implement the classes shown below (as well as the exact member functions that are listed).
Note: It is okay to add additional functions or variables as desired. **You cannot add extra parameters to the functions that are listed.**

```
class Node {
public:
   int val; // the value that this node stores
   Node *next; // a pointer to the next node in the list
   // you can add constructors or other functionality if you find it useful or
necessary
};
```

Note: `Node` is being used akin to a struct (with public member variables). This is intentional so that you can easily modify the member variables from within the `Linked_List` class.

```
class Linked_List {
private:
   int length; // the number of nodes contained in the list
   Node *head; // a pointer to the first node in the list
   // anything else you need...
public:
   int get_length();
   // note: there is no set_length(int) (the reasoning should be intuitive)
   void print(); // output a list of all integers contained within the list
   void clear(); // delete the entire list (remove all nodes and reset length to 0)
   void push_front(int); // insert a new value at the front of the list
   void pop_front(); // remove the node at the front of the list


   // you can add extra member variables or functions as desired
};
```

Show your completed work and answers to the TAs for credit. You will not get points if you do not get checked off!

Submit your work to TEACH for our records **(Note: you will not get points if you don't get checked off with a TA!!!)**

1. Create a **zip file** that contains all files you've created in this lab:
2. Transfer the tar file from the ENGR server to your local laptop.
3. Go to TEACH.
4. In the menu on the right side, go to **Class Tools → Submit Assignment**.
5. Select **CS162 Lab10** from the list of assignments and click "**SUBMIT NOW**"
6. Select your files and click the Submit button.

That's it! You've completed all labs of the course! Congratulations! We wish you all the best in the future :)