

CS 240: Data Structure

Problem Set 1

Due: 02/09/2018

Instructions:

Please follow the instruction given at each problem for submission. The deadline of each assignment is 11:59 pm on the due date, unless otherwise specified. It is important that your code/solution is straight forward, not cumbersome. At the beginning of each cpp file, you must include the following:

- 1. Description:** description of the program (later we will elaborate this more, but for now, just describe what the program does)
- 2. Author:** the person who writes this program
- 3. Date:** the very last date/time the program is modified

First Name:

Last Name:

Score: /100+10

Problem 1 Software Development: Initial Stage: 40pts

In chapter 1, we briefly discussed about software development. There are various types of development models, such as Water Fall and Agile. Let us imagine you are to launch a new project that will help all offline stores (globally) to sell online. Let us call this project PelicanBlock. Let us assume most offline stores do not have online pages and have no clue about arranging the delivery. You want your project to soar, please briefly describe how you would attack this problem (based on the 2nd slide of chapter 1: Problem analysis and specification, Design, Coding, Testing/Execution/Debugging and Maintenance) and write up your 2-page mini white paper to attract potential customers and investors. You can even design your own currency, let say PelicanToken (PLT). Let this document be PelicanBlockWhitePaperDraft.doc (or PelicanBlockWhitePaperDraft.pdf).

Problem 2 Programming: Validator + Perm: 5 +15 pts

In class, we learned how to write the factorial function and the combinatorial function. Remember that $n! = n \times (n - 1) \times \cdots \times 1$ and $C(n, k) = n! / ((k!) \times (n - k)!)$. Please write out the permutation function **Perm (int n, int k)** where $Perm(n, k) = n! / k!$. Your program will ask for inputs in the main function. I might input negative integers or k is greater than n. Your code in main (or you can write a short checking function) will verify the inputs. If not valid, your main will print out “Invalid inputs” and continue to ask for inputs till valid inputs are obtained. Once the inputs are valid, then your main will call the Perm function to compute then the main will print out the result. Please name this file **perm.cpp**.

Problem 3 Programming: Struct + Sorting : 40 pts

We simply talked about the structure LinkedList (only one direction, not Double LinkedList). Let us define the structure of the LinkedList as

```
struct node
{int value;
node * nxt;
};
```

The exercise practices on (1) struct, pointers, dereferencing and (2) of course, a brute-force type of algorithm implementation. So, in your main program, I will embed an array of integers, let say `int test[] = { ... }` (I will decide what numbers to put in when testing; let us assume the array size is 20).

Your main will iterate through the elements (each will be stored in a new node later) in the test array and store them into the LinkedList properly. You will need to write (a) an insert function `node* insert(node *p, int k)` where p is the address of the beginning of the LinkedList you have built so far and k is the integer you just read in from the array. Insert function takes the LinkedList and checks where to insert the new node (containing k) into the list such that the values in the nodes are in an increasing order and (b) a print function `void printLL(node *p)` that prints out the values inside the LinkedList from the root to the end.

In this example you are working on a new data structure and a simple sorting algorithm. Let us name this file as **linkedlist.cpp**.

Bonus: 10 pts: Same questions as above, but now the content (value) of the node can be all integers or all strings (exclusively) since your clients have not decided what data type to use inside the node. What can you do to attack this problem if you are really pressed for time because the product integration test is next week? Please simply modify your `linkedlist.cpp` and call this modified file **opentypell.cpp**.