MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

CNG213 C Programming – Programming Assignment 1

Date handed out: **Wednesday 04 November 2015**

Date submission due: **Monday 16 November 2015**

Introduction

This assignment aims to help you practice linked list data structure and basic linked list operations. Your main task in this assignment is to create a small application for Microbiologists which at the backend uses linked list data structure and basic linked list operations.

Programming Use Case

Microbiologists estimating the number of bacteria in a sample that contains bacteria that doe not grow well on solid media may use a statistical technique called the most probable number (MPN) method. Each of five tubes of nutrient medium receives 10 ml of the sample. A second set of five tubes receives 1 ml of sample per tube, and in each of a third set of five tubes, only 0.1-ml of sample is placed. Each tube in which bacterial growth is observed is recorded as a positive, and the number for the three groups are combined to create a triplet such as 5-2-1, which means that all five tubes receiving 10 ml of sample show bacterial growth, only two tubes in the 1-ml group show growth, and only one of the 0.1-ml group is positive. A microbiologist would use this combination-of-positives triplet as an index to a table like the one below.

Table-1: Bacterial Concentrations for Most Probable Number Method

Combination of Positives	**MPN Index/100 ml**	**95% Confidence Limits**	
		Lower	**Upper**
4-2-0	22	9	56
4-2-1	26	12	65
4-3-0	27	12	67
4-3-1	33	15	77
4-4-0	34	16	80
5-0-0	23	9	86
5-0-1	30	10	110
5-0-2	40	20	140
5-1-0	30	10	120
5-1-1	50	20	150
5-1-2	60	30	180
5-2-0	50	20	170
5-2-1	70	30	210
5-2-2	90	40	250
5-3-0	80	30	250
5-3-1	110	40	300

Below to determine that the most probable number of bacteria per 100 ml of the sample is 70, and 95% of the samples yielding this triplet contain between 30 and 210 bacteria per 100 ml.

Programming Requirements

Define a structure type to represent one row of the MPN table. The structure will include one String component for the combination-of–positives triplet and three integer components in which to store the associated most probable number and the lower and upper bounds of the 95% confidence range. Write a program to implement the following algorithm for generating explanations of combination-of-positive triplets.

a. Load the MPN table from a file into a linked list called mpn_table.
b. Allow user to enter a new data into this linked list which means entering data given in a new row like this:

Combination of Positives	**MPN Index/100 ml**	**95% Confidence Limits**	
		Lower	**Upper**
5-3-2	140	60	360

   Each of these items will be entered separately by the user.
c. Repeatedly get from the user a combination-of-positives triplet, search for it in the combination-of-positives components of mpn_table, and then generate a message such as:
   For 5-2-1, MPN = 70; 95% of samples contain between 30 and 210 bacteria/ml.

d. Define and call the following functions.

Load_Mpn_Table -- **Takes as a parameter the name of the input file**. This function opens the file, fills the mpn_table linked list with the data from this file, and closes the file. Dynamic memory allocation and population should be done inside the function. You should not make any assumption about the size of the data. You can assume that the data is sorted in the file based on the combination of positives. **Then it returns the linked list as the function result**.

Display_Mpn_Table – **Takes the mpn_table linked list as a parameter** and displays the table on the screen with appropriate messages. If the table has not been loaded then appropriate error message should be displayed to the user, as well. **This function will not return anything.**

Insert_Mpn_Table: -- This function takes the mpn_table linked list and collects data from the user and inserts it into the linked list. You need to pay attention that the table is sorted based on the combination of positives. Therefore, the new record should be inserted into the right location to keep the data sorted. **This function will not return anything.**

Search_Mpn_Table -- **Takes as parameters the mpn_table linked list (which possibly includes its actual size), and a target string representing a combination–of-positives triplet**. **Returns the position of the structure whose combination-of-positives component matches the target or -1 if not found.**

Main function that coordinates these functions and extra functions needed. For each extra function, specify the input and output parameters. You need to make sure that your function prototypes are clearly defined.

Please follow the modular programming approach. In C we use functions referred to modules to perform specific tasks that are determined/guided by the solution. Remember the following tips!

- **Modules can be written and tested separately!**
- **Modules can be reused!**
- **Large projects can be developed in parallel by using modules!**
- **Modules can reduce the length of the program!**
- **Modules can also make your code more readable!**

Grading:
Your program will be graded as follows:

Grading Point	**Mark (100)**
Load_Mpn_Table	25 points
Display_Mpn_Table	20 points
Insert_Mpn_Table	20 points
Search_Mpn_Table	20 points
Main function	15 points

NOTE: Remember to have good programming style (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.). Each of the items above will include10% for good programming style.