Lab #3
Memory, Arrays, Pointers, Strings, Structures

Purpose:

The purpose of this lab is to write some code in C that uses aspects of the language that has been covered in class thus far.

Source code files for the lab are on Canvas. You will save all your source files for the lab to the GitHub account you created in Lab 0.

Preparation:

Review class presentations and chapters from the text according to the syllabus.

Procedure:

1) Start up CodeLite and open the workspace ECEGR2020 you created for Lab 0
2) In the workspace, create a new project: File -> New -> New Project. Name the project Lab_3 and make the Type a “Simple executable (gcc)”
3) For each function below, program, run and debug each. In your lab report, be sure to place the output of each function to show that your program worked.

Program the Following:

A) Create a function: void ReverseArray(void)

   With the following global array of ints: int theArray[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

   In the function, reverse the elements of the array and then print out the array contents

B) Using this structure:

```
struct student
{
   int  ID;
   char firstName[30];
   char lastName[30];
   float GPA;
} Student;
```

Write a function that allocates a Student structure on the heap and fills in each member of the structure with input from the user.

Have the user input the number of students in a class

Allocate an array of Student pointers to hold the number of students in the class

For each student in the class, allocate a Student structure and have the user fill in the information

After the user has entered the info for each student in the class, sort the Students by GPA, largest to smallest (hint – use the fact that you have an array of pointers when sorting)

After sorting, save the number of students and all the Student information to a file

Make sure to clean up the memory that was allocated before exiting the program

Check in the file to GitHub along with your source code

C) Write a function that that reads the student file and re-creates the list of students in the class.

Print out the student information, the number of students and the average GPA of the class

D) Write a main program that performs the following steps:

- Prompt the user to enter a string, and let them type it in. This could be an entire sentence, with the newline indicating the end of the string. You may assume the string will be no more than 100 characters, so declare your array accordingly.

- Display the following menu:

  A) Count the number of vowels in the string
  B) Count the number of consonants in the string
  C) Convert the string to uppercase
  D) Convert the string to lowercase
  E) Display the current string
  F) Enter another string

  M) Display this menu
  X) Exit the program

- Enter a loop, allowing the user to type in a menu choice each time. Loop should continue until the user enters the command to exit. Upper and lowercase letters should be allowed for the menu choices.
- When the A or B commands are entered (counting vowels or consonants), call the corresponding function, then print the result
- When the C or D commands are chosen, just call the appropriate function to convert the string. Do not do any output from main on these commands.
- When E is chosen, print the contents of the stored string.
- When F is chosen, allow a new string to be typed -- this will replace the previous one.
- The menu should only be displayed once at the start, and then again whenever the M option is selected

E) Create a new option for the program from (D)

R) Reverse words in string

This option should reverse each of the words in the string

Example: cat dog mouse => tac god esuom