

DATA STRUCTURES & ALGORITHMS 1

BATCH – B

[FRIDAY MARCH 15, 2019: 3:30 PM – 6:30 PM]

LAB ASSIGNMENT – 8

CODE:assign08

NOTES:

1. Please carefully read all assignments and there is **no choice**.
 2. Use the same files names used in the tutorials.
 - a. Copy the three project folders into a master folder.
 - b. Zip/Tar the master folder and submit it.
 - c. Please name the master folder ‘master_assign_08_XXX’ where XXX is the last three digits of your rollno.
 3. **Follow variable and function naming conventions**
 - a. except for variables in for-loop, none of the other variables should be a single character.
 - b. The variable names and function names should indicate what they are storing/computing. For this assignment, we have given you some of the variable names and function names to use. They are highlighted as **function_name** or **variable_name**
 - c. All global variable should start with ‘g_’
 4. Indentation improves readability. Please pick an indentation style and **indent your code** appropriately.
 5. Follow constants and type naming
 - a. All **constants** should be defined using **IFDEF** and **DEFINE**
 - b. All **structures** should have a **TYPDEF** to a simpler name
 6. When in doubt about naming or style conventions, consult the following link:
<https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html>
-

PROBLEMS [Total Marks: 20]:

Today we are going to practice how to write modular C code. So, far we have created one C file which contains: global variables and function declarations, type and constant declarations and a main function. The primary objective of today is to learn how to split the interface and implementation into separate files and to manage all the source code as a single project, with the help of make files.

Exercises:

1. [6 Marks] Familiarize yourself with the simple makefile tutorial in the link:
<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor>

- a. Follow along the tutorial to create a project with the code provided in the tutorial
 - b. Experiment with different versions of make files
(Warning: You will be asked to explain the meaning and purpose of the different flags and macros)
2. [6 Marks] Familiarize yourself with the modules and compilation tutorial in the link:
<https://www.cs.bu.edu/teaching/c/separate-compilation/>
 - a. Follow along the tutorial to create a project with the code provided in the tutorial
 - b. Pay special attention to problems highlighted in the tutorial including multiple inclusion problems (Now you can truly appreciate IFNDEF)
3. [8 Marks] Implement a simple linked list for student structure (rollno, name, cgpa). The linked list should support the following operations:
 - a. add(node) → add the node to the end of the list
 - b. findrollno(rollno) → find and return the node if it exists
 - c. print(node) → print details of a single node
 - d. printfrom(node) → prints the current node using 'print' function you wrote and recursively prints till the end of the list
 - e. printall() → uses printfrom() and print() to print the entire linked list
 - f. delete_by_rollno(rollno) → finds the node with the given rollno and deletes it(Hint: You can assume that roll numbers are unique and need not verify for its uniqueness)

Note:

1. penalties for violating style and naming conventions are doubled for today
2. Please make sure that you have atleast one file which contains *almost* nothing but main.