DATA STRUCTURES & ALGORITHMS 1

BATCH – A [WEDNESDAY FEBRUARY 20, 2019: 3:30 PM – 6:30 PM]

LAB ASSIGNMENT – 5 CODE:assign05

NOTES:

1. Please carefully read all assignments and there is no choice.
2. **Use the template for this assignment**
3. Each problem in this assignment has to be answered in the same c file.
4. Create a .c file following the file name convention:
    a. If your roll number is 'abc' and assignment code is 'assignXX'. Then use the following file name convention as follows: 'abc-assignXX.c'
    b. For example, if the roll number is 92 and assignment code is assign01, then the file name should be 092-assign01.c
    c. Strictly follow the file name convention. When you are ready, submit the solution via google classroom.
5. Follow 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

---

Create a Structure **'student'** with the following details

1. Student name
2. Rollno (int is fine)
3. CGPA
4. A pointer to favourite 'student' (struct) instance (we will use this pointer to point to the student's favorite student)

PROBLEMS [Total Marks: 20]:

1. [Marks: 5] Do the following
    a. Use 'ifndef' to 'define' a constant TOTAL_STUDENTS with value 5. From this point onwards you should only use this constant in your program.
    b. Use 'Typedef' to the 'structure student' as 'Student', to make it easier for you to use in other places. From this point onwards you should only use 'Student' in your program.
    c. Declare a global array 'g_students' to store **an array of pointers** to 'Student'

2. [Marks: 5] Write a function 'populate_student_details' for populating the 'g_students' array with values provided by the user (via scanf). Make sure the memory for Student instances are allocated in **the Heap** (use malloc). Initialize all favorite_student pointer to NULL.

3. [Marks: 5] Do the following:
    a. Write a function 'find_index' which takes as input a roll_no and finds the student with that roll_no in g_students. The function returns the index (position) in the array as the output.
    b. Write a function 'make_favorite_student' which takes as input two roll numbers (say roll1 and roll2). The function makes the second student (roll2) the favorite of the first_student. You must use the 'find_index' function for this.

4. [Marks: 5] Do the following:
    a. Write a **recursive** function 'print_fav_chain' which takes as input a Student pointer (say: sptr). It prints the details of the student pointed by 'sptr' and then proceeds to print the details of the favorite student (via recursion).
    b. Write a function 'print_from_student' which takes as input the roll_no (int). It uses the find_index to find the student corresponding to the roll_no. Then it invokes 'print_fav_chain' to print the chain of favorites.
