Course Title: Compiler Design Sessional; Course Code: CSE 360; Credit: 0.75

Assignment

1. Write a C/C++ program that reads text from a file and prints on the terminal each input line, *preceded by the line number*. The output will look like -

   ```
   1    This is the first trial line in the file,
   2    and this is the second line.
   ```

   Try the problem once using fgetc() function and once using fgets() function for reading the input. Why is fread() not suitable for this purpose?

   Do not ignore the value returned by the functions fgetc() and fgets(). After this exercise the you should be comfortable with the formatted input and output functions of C.

2. Write a C/C++ program to identify whether a given input line is a **comment** or not.

3. Write a C/C++ program to test whether a given **identifier** is valid or not.

4. Write a C/C++ program that reads text from a file, then count and prints the number of **character** exist in the inputted text file.

5. Write a C/C++ program that reads text from a file, then count and prints the number of **line** exist in the inputted text file.

6. Write a C/C++ program that reads text from a file, then **delete** the existing **comment** and save the output text file (without comment). Also count and print the number of deletion.

7. Write a C/C++ program that reads text from a file, then **delete** the existing **tabs (spaces)** and **new line** and save the output text file. Also count and print the number of deletion.

8. Imagine the syntax of a programming language construct such as *while-loop* --

   while (*condition*)
   begin
   statement ;
   :
   end

   where *while*, *begin*, *end* are keywords; *condition* can be a single comparision expression (such as *x == 10*, etc.); and *statement* is the assignment to a location the result of a single arithmatic operation (eg., *a = 5 * b*).

   Write a C/C++ program that verifies whether the input follows the above syntax.

***** Write an efficient C/C++ program (by pen and paper) for solving each of the above algorithm (problem) obeying the general guidelines given below (i-iii):**

i. Show user-friendly messages to make the program interactive.
ii. Use symbolic constant as per your requirements.
iii. Try to avoid static initializations.

Finally submit your solution as scanned copy (pictures) in a single pdf file with mentioning your name, student id and course code.