# CSL411 COMPILER DESIGN

## CYCLE I

1. Design and implement a lexical analyzer using C language to recognize all valid tokens in the input program. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments
2. Write a program to find First and Follow of any given grammar.
3. Design and implement a recursive descent parser for the grammar given below.

   E → TE’

   E’→ +TE’/$

   T→ FT’

   T’→ *FT’/$

   F→ (E)/ID
4. Construct a Shift Reduce Parser for a given language.
5. Write a program to perform constant propagation.
6. Implement Intermediate code generation for simple expressions in quadruple, triple, indirect triple representation.
7. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.