**C7A-(22-26)**

**Cycle I (submit by 2 /7/2025 - 17/7/2025)**

- Design and implement a lexical analyzer for given language using C and the lexical analyzer should ignore redundant spaces, tabs and newlines.
- Write a program to find ε – closure of all states of any given NFA with ε transition.
- Write a program to convert NFA with ε transition to NFA without ε transition.
- Write a program to convert NFA to DFA
- Write a program to minimize any given DFA.

**Cycle II (submit by 23/7/2025 -6/8/2025)**

- Write a lex program to recognize all strings which does not contain first four characters of your name as a substring.
- Write a YACC program to recognize a valid variable which starts with a letter followed by any number of letters or digits.
- Implementation of Calculator using LEX and YACC
- Convert the BNF rules into YACC form and write code to generate abstract syntax tree
- Write a YACC program to check the syntax of FOR statement in C

**Cycle III (Submit by 7/8/2025 – 17/9/2025)**

- Develop an operator precedence parser for a given language.

- Write a program to Simulate First and Follow of any given grammar.
- Construct a recursive descent parser for an expression.
- Construct a Shift Reduce Parser for a given language.

**Cycle IV (submit by 18/9/2025 – 9/10/2025)**

- Implement Intermediate code generation for simple expressions.
- 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.