List of Experiments:

1. Environment setup and run “Hello World” program
2. Implement the lexical analyzer Flex or Lex to match a string:
    a. ([A-Z][a-z]+)\s([a-z0-9]+)
    b. [A-Z]\w+\s\w+\s\w+[!]
    c. (\w+ats?\s)+
3. A Lex program to identify an integer number
4. A Lex program to identify a Teletalk number
5. A Lex program to identify a floating point number
6. A Lex program to recognize and count the number of identifier in a given input
7. A Lex program to count the characters, words, spaces, and lines in a given input
8. A Lex program to identify exponential numbers
9. A Lex program to identify “to be” verb
10. A Lex program to identify a complex number
11. A Lex program to recognize whether a given sentence is simple, compound or complex
12. Implement Yacc program to evaluate an expression (Calculator)
13. Implement Yacc program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits
14. Implement Yacc program to recognize a valid arithmetic expression that uses operators +, -, * and /
15. Implement Yacc program to recognize strings ‘aaab’, ‘abbb’, ‘ab’ and ‘a’ using the grammar (a^nb^n, n>0)
16. Write a YACC program to recognize strings of the form $a^nb^{n+m}c^m$, n,m>=0.
17. Write a LEX program to count the number of comment lines in a C Program. Also eliminate them and copy that program into a separate file.
18. Write a YACC program to recognize a nested (minimum 3 levels) FOR loop statement for C language.
19. Write a LEX program to recognize and count the number of identifiers, operators and keywords in a given input file.