|  |  |
| --- | --- |
| 1 | Implement lexical analyzer for subset of English language using LEX. |
| 1 | Implement lexical analyzer for subset of 'C' language using LEX. |
| 1 | Implement lexical analyzer for subset of english language using LEX. Input filename as command line argument |
| 1 | Implement lexical analyzer for subset of 'C' language using LEX. Input filename as command line argument |
| 1 | Implement word count program using LEX. |
| 1 | Implement word count program using LEX. Input filename as command line argument. |
| 2 | Implement lexical analyzer for subset of english language using LEX. Build symbol table to dynamically declare and lookup parts of speech. |
| 2 | Implement a lexical analyzer to input 'C' program file and a)Count number of comments b) Eliminate comments and c) Store output in another file |
| 2 | Implement a lexical analyzer to input 'C' program file and a)Count number of simple and compound statements . Input filename as command line argument. |
| 3 | Write a YACC specification to implement arithmetic calculator. |
| 3 | Write a YACC specification to implement scientific calculator. |
| 3 | Write a YACC specification to implement calculator.Extend to handle variables with single letter names. |
| 4 | Write a YACC specification to check syntax of "for" statement of 'C' language. |
| 4 | Write a YACC specification to check syntax of "switch… case" statement of 'C' language. |
| 4 | Write a YACC specification to check the syntax of "if" and "if … else" statements of 'C' language. |
| 4 | Write a YACC specification to recognize subset ofenglish language sentences. |
| 5 | Generate intermediate code for input "C" language assignment statement. |
| 5 | Generate intermediate code for input "C" language if ..else assignment statement. |
| 6 | Generate appropriate target code for given intermediate code assuming suitable processor details. |
| 6 | Implement register allocation algorithm on input code. |
| 7 | Implement Common sub expression elimination optimization on input 3-address code block. |
| 7 | Implement constant folding optimization on input 3-address code block. |
| 7 | Implement " loop invariant computation" optimization on input 3-address code block. |

Program to count number of identifiers in a given input file

Program to count number of scanf and printf statement in a “C” program & replace them with readf and writef statements.

Program to count number of comment lines in a given C program. Also eliminate them and copy that program into separate file.

Program to count number of

(1) Positive and negative integers

(2) Positive and negative fractions

Program to count number of vowels and consonants in a given string.