Compiler design lab work list

1. *Write a program to implement lexical analyser by using c program.*
2. *Write a program to convert an expression from Infix to Postfix.*
3. *Write a program to find the FIRST of a given program.*
4. *Write a program to find the FOLLOW of a given program.*
5. *Write a program to implement Recursive Descent Parser of a given grammar.*
6. *Write a program to implement Shift reduce parser.*
7. Write a program to implement of quadruples of a given expression.
8. Construction of state transition table for NFA with epsilon moves
9. Construct a DFA from a given regular expression
10. Write a program for LL(1) parser of a given grammar
11. *Write a C program for implementating shift reduce parsing using a given grammar.*

**Write a program to implement lexical analyser by using c program**

<https://www.geeksforgeeks.org/c-program-detect-tokens-c-program/>

<http://karthikalapati.blogspot.com/2016/04/c-program-to-implement-lexical-analyzer.html>

<https://www.thecrazyprogrammer.com/2017/02/lexical-analyzer-in-c.html>

**Write a program to convert an expression from Infix to Postfix.**

<https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/>

<https://www.technicalseek.com/c-solved-programs/program-to-convert-infix-to-postfix-using-stack/>

<http://interactivepython.org/runestone/static/pythonds/BasicDS/InfixPrefixandPostfixExpressions.html>

**Write a program to find the FIRST of a given program**

[**http://karthikalapati.blogspot.com/2016/04/c-program-to-implement-first-of-given.html**](http://karthikalapati.blogspot.com/2016/04/c-program-to-implement-first-of-given.html)

1. Each Non terminal character is represented by one Uppercase letter.
2. Each Terminal character is represented by one lowercase letter.
3. LHS and RHS of each production will be separated by a "=" symbol.
4. There will be no Blank space within a production string.
5. Assumed that Left recursion is avoided in all input productions

Epsilon is denoted by #

<https://www.geeksforgeeks.org/program-calculate-first-follow-sets-given-grammar/>

The rules for finding FIRST of a given grammar is:

1. If X is terminal, FIRST(X) = {X}.
2. If X → ε is a production, then add ε to FIRST(X).
3. If X is a non-terminal, and X → Y1 Y2 … Yk is a production, and ε is in all of FIRST(Y1), …, FIRST(Yk), then add ε to FIRST(X).
4. If X is a non-terminal, and X → Y1 Y2 … Yk is a production, then add a to FIRST(X) if for some i, a is in FIRST(Yi), and ε is in all of FIRST(Y1), …, FIRST(Yi-1).

**Write a program to find the FOLLOW of a given program**

[**http://karthikalapati.blogspot.com/2016/04/first-and-follow-of-given-grammar-using.html**](http://karthikalapati.blogspot.com/2016/04/first-and-follow-of-given-grammar-using.html)

[**http://crazywebappdeveloper.blogspot.com/2013/10/follow-set-of-given-grammar-using-c.html**](http://crazywebappdeveloper.blogspot.com/2013/10/follow-set-of-given-grammar-using-c.html)

1. Write a program to implement Shift reduce parser.

[**https://www.geekysplash.com/compiler-design-shift-reducing-parsing-c/?fbclid=IwAR0b8PDBuqN5\_DFQWIXzU4acJoWZy8FZWay8qd7FkLTQB\_lxo7JJonv64q0**](https://www.geekysplash.com/compiler-design-shift-reducing-parsing-c/?fbclid=IwAR0b8PDBuqN5_DFQWIXzU4acJoWZy8FZWay8qd7FkLTQB_lxo7JJonv64q0)

[**http://3rdyearcselabprograms.blogspot.com/2010/01/shift-reduce-parser\_15.html**](http://3rdyearcselabprograms.blogspot.com/2010/01/shift-reduce-parser_15.html)

1. Construct a DFA from a given regular expression

[**https://www.quora.com/Can-we-write-a-C-program-to-convert-regular-expression-to-DFA**](https://www.quora.com/Can-we-write-a-C-program-to-convert-regular-expression-to-DFA)

Write a program to implement of quadruples of a given expression.

<https://vipinnpillai.blogspot.com/2011/11/quadruple-c-code_20.html>

<http://3rdyearcselabprograms.blogspot.com/2010/01/lexical-analyser-in-c.html>

<http://3rdyearcselabprograms.blogspot.com/2010/01/code-generation.html>