# COMPILER DESIGN LAB EXAM
JAN 2020-21

Name: Abhay V Ashokan
S7, CSE
Roll No: 2

AIM: Implement a syntax analyser in yacc for structure declaration statements in C.

## Algorithm

step1 : start

step 2 : Read the name of the program to be analysed as input.

step 3 : If the file does not exist, or you do not have the permission to open the file, display "Invalid file name" and exit.

Step 4 : In the lex file parse each character and check whether pattern (P) matches any valid token, initialize lineCounter as 1.

4.1 If P is 'int', 'char', 'float', 'double', 'long', 'char*', 'int*', 'float*', 'long*' return TYPE

4.2 If p is "struct" return STRUCT

4.3 If p matches regular expression of an identifier, return ID

4.4. For new line, increment lineCounter by 1.

4.5. For all other cases, return the character (token).

Step 5: Check whether the file ~~and~~ contents match the following context free grammar.

STRUCTURE → STRUCT ID BLOCK ; |
STRUCT ID BLOCK ID

BLOCK → { CODE }

CODE → STATEMENT CODE | STATEMENT;

STATEMENT → TYPE IDLIST ;

IDLIST → ID, IDLIST | ID

Step 6: If the context free grammar is satisfied, print "Valid ~~expression~~ Structure declaration"

Step 7: If ~~it~~ an ~~or~~ error occurs, display the error message with line number.

Step 8: STOP

OUTPUT

Enter file name of the program: structure.c
Valid structure Declaration.

Readme

- Compile program using the command
  lex exam.l && yacc exam.yacc -d
- Execute using the command
  gcc lex.yy.c y.tab.c -ll && ./a.out
- Enter file name

- The message will be obtained as output.

RESULT : Successfully implemented program to implement a syntax analyzer in yacc for structure statements in C.