33304

Assignment - 11

* Problem Statement :- Write a program to implement a calculator using lex and yacc tools.

* Objective :- To understand the process of lexical analysis and tools such as lex and Yacc.

* Theory :-

- Lexical Analysis → The action of scanning the source program into proper syntactic classes. It scans the program and converts it into basic elements or tokens of the language.

The goal is to check the validity of a source string and to determine its syntactic structure. For an invalid string the parser issues diagnostic messages reporting the cause and nature of errors.

For a valid string, it builds a parse tree to reflect sequence of derivations or reduction performed.

- Bottom up Parsing → A bottom up parser constructs a parse tree for a source string through a sequence of reductions. The source string is valid if it ~~is~~ can be reduced to 'S', the distinguished symbol of grammar 'G'. If not, an error is reported and indicated during the process of reduction

- Top down Parsing → The top down parsing according to grammar 'G' tries to derive the matching source string through the sequence of derivations, starting with distinguished symbol of G.

33304

- **Shift reduce parser →** This method is bottom up parsing → which attempts to build the parse for an input string beginning at the leaves (bottom) and working up towards the root (up). In this process a string w is reduced to the start symbol of the grammar. A

At each step a string matching the right side of the production is replaced by the symbol on the left. Each one of this replacement step is called as a reduction.

- **Input structure to lex →**

```
... Definition section ...
          %%
   ... Rules section ....
          %%
   ... Code section ...
```

Ex:-

```
%%
    ECHO ;
\n  ECHO;
%%
int yywrap (void) {
        return 1;
}
int main (void) {
        yylex();
        return 0;
}
```

Echo is an action and predefined macro in lex that writes code matched by the pattern.

33304

- Overall process of lex compilation →

lex source prog → lex compiler → lex.yy.c

lex.yy.c → c compiler → a.out

Input stream → a.out → sequence of tokens
input.c

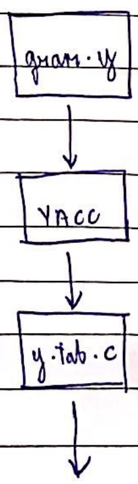
- YACC →

1. YACC stands for 'Yet Another Compiler Compiler"
2. It reads the grammar and generates C code for a parser.
3. Grammar written Backus Naur Form (BNF)
4. BNF grammar used to express content - free languages.
5. Ex :- To parse an operation, do reverse operation (reducing the expression)
6. This is known as bottom - up or shift - reducing parsing
7. It uses stack for storing (LIFO).

- Sections of YACC input →

~~%~~ ... definitions ...
%%
... rules ....
%%
... subroutines ...

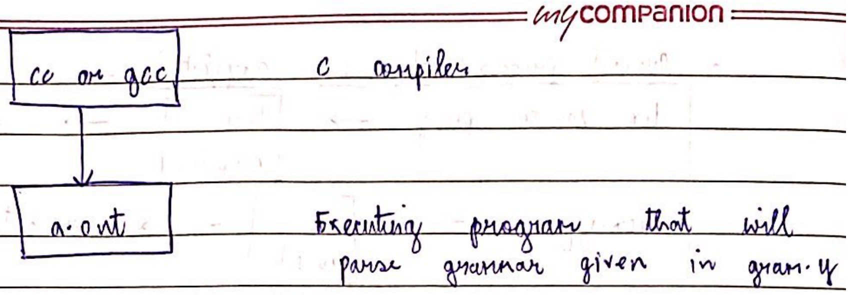
- Basic operational sequence of YACC →



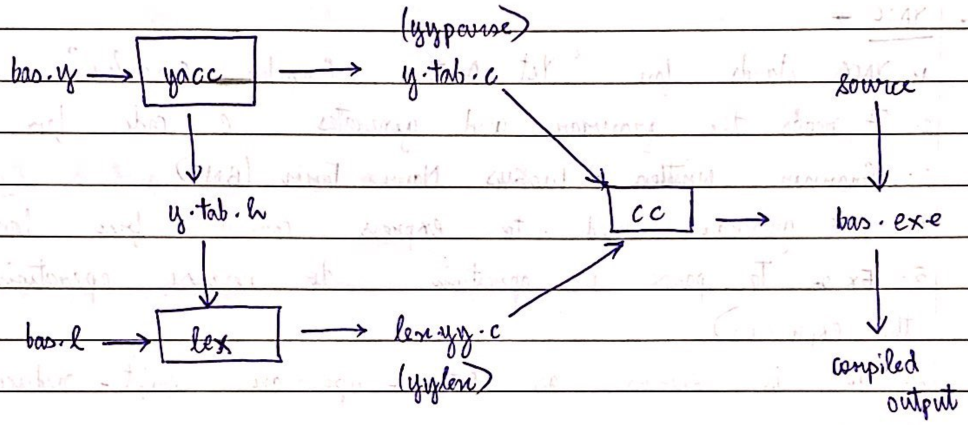
The file containing desired grammar in YACC format

YACC program

C source program created by YACC



- Linking of lex and YACC→



* Conclusion :- Thus we have studied about lex and YACC, their structure, linkage and implement a calculator using lex and yacc.