**PRACTICAL-7**

**AIM : To study about LEX Parser.**

**INTRODUCTION:**

* Lex is a program that generates lexical analyzer. It is used with YACC parser generator.
* The lexical analyzer is a program that transforms an input stream into a sequence of tokens.
* It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.

Function of lex is as follows:

* Firstly lexical analyzer creates a program lex.1 in the Lex language. Then Lex compiler runs the lex.1 program and produces a C program lex.yy.c.
* Finally C compiler runs the lex.yy.c program and produces an object program a.out.
* a.out is lexical analyzer that transforms an input stream into a sequence of tokens.

LEX File Format:

* A Lex program is separated into three sections by %% delimiters. The formal of Lex source is as follows:

{ definitions }

%%

{ rules }

%%

    { user subroutines }

* **Definitions** include declarations of constant, variable and regular definitions.
* **Rules** define the statement of form p1 {action1} p2 {action2}....pn {action} Where **pi** describes the regular expression and **action1** describes the actions what action the lexical analyzer should take when pattern pi matches a lexeme.
* **User subroutines** are auxiliary procedures needed by the actions. The subroutine can be loaded with the lexical analyzer and compiled separately.

**EXAMPLES**:

1. Write a program to count vowels and consonants in given entered string.

%{

    int vow\_count=0;

    int const\_count =0;

%}

%%

[aeiouAEIOU] {vow\_count++;}

[a-zA-Z] {const\_count++;}

%%

int yywrap(){}

int main()

{

    printf("Enter the string of vowels and consonents:");

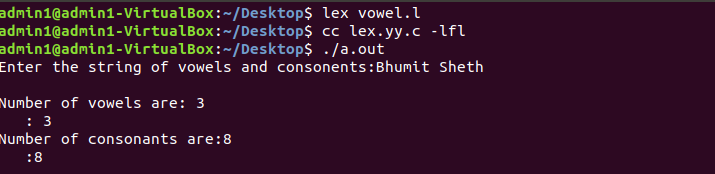
    yylex();

    printf("Number of vowels are:  %d\n", vow\_count);

    printf("Number of consonants are:  %d\n", const\_count);

    return 0;

}



1. Write a program to count number of spaces, lines and tabs.

%{

#include<stdio.h>

int lc=0, sc=0, tc=0, ch=0; /\*Global variables\*/

%}

/\*Rule Section\*/

%%

\n lc++; //line counter

([ ])+ sc++; //space counter

\t tc++; //tab counter

. ch++;     //characters counter

%%

main()

{

    // The function that starts the analysis

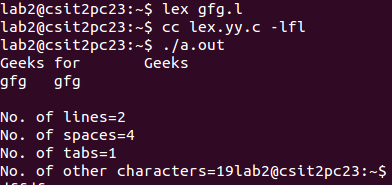
    yylex();

    printf("\nNo. of lines=%d, lc);

    printf("\nNo. of spaces=%d, sc);

    printf("\nNo. of tabs=%d, tc);

    printf("\nNo. of other characters=%d, ch); }



**CONCLUSION**:

Thus, we can conclude that Lex is a tool known for Lexical Analysis . It’s main job is to break up an input stream into more usable elements called as tokens. It uses regular expression matching; typically to ‘tokenise’ the contents of the file.