**Ex.No:**

**LEXICAL ANALYZER USING Lex Tool**

**Aim:**

To write a lex program to implement Lexical Analyzer to identify the patterns like identifiers, keywords, comments, operators and constants.

**Description:**

Lex is a software tool used for writing Lexical Analyzers. Lex source is a specification of tokens in the form of a list of regular expressions together with an action for each regular expression. The action pass an indication of the token found to the parser and makes an entry in the symbol table. The Output program generated by the LEX compiler is the finite automata.

+-------+

Source 🡪| Lex | 🡪yylex

Input 🡪 | yylex | 🡪 Output

+-------+

The general format of Lex source is:

{definitions}

%%

{rules}

%%

{user subroutines}

where the definitions and the user subroutines are often omitted. The second %% is optional, but the first is required to mark the beginning of the rules. The absolute minimum Lex program is thus

%%

**/\*Lexical Analyzer using Lex Tool\*/**

%option noyywrap

letter [a-zA-Z]

digit [0-9]

id [\_|a-zA-Z]

AO [+|-|/|%|\*]

RO [<|>|<=|>=|==]

pp [#]

%{

int n=0;

%}

%%

"void" printf("%s return type\n",yytext);

{letter}\*[(][)] printf("%s Function\n",yytext);

"int"|"float"|"if"|"else" printf("%s keywords\n",yytext);

"printf" printf("%s keywords\n",yytext);

{id}({id}|{digit})\* printf("%s Identifier\n",yytext);

{digit}{digit}\* printf("%d Numbers\n",yytext);

{AO} printf("%s Arithmetic Operators\n",yytext);

{RO} printf("%s Relational Operators\n",yytext);

{pp}{letter}\*[<]{letter}\*[.]{letter}[>] printf("%s processor

Directive\n",yytext);

[\n] n++;

"."|","|"}"|"{"|";" printf("%s others\n",yytext);

%%

int main()

{

yyin=fopen("sample.c","r");

yylex();

printf("No of Lines %d\n",n);

}

**/\*Input\*/**

#include<conio.h>

void main()

{

int a,b,c;

}

**/\*Output\*/**

#include<conio.h> processor Directive

void return type

main() Function

{ others

int keywords

a Identifier

, others

b Identifier

, others

c Identifier

; others

} others

No of Lines 5