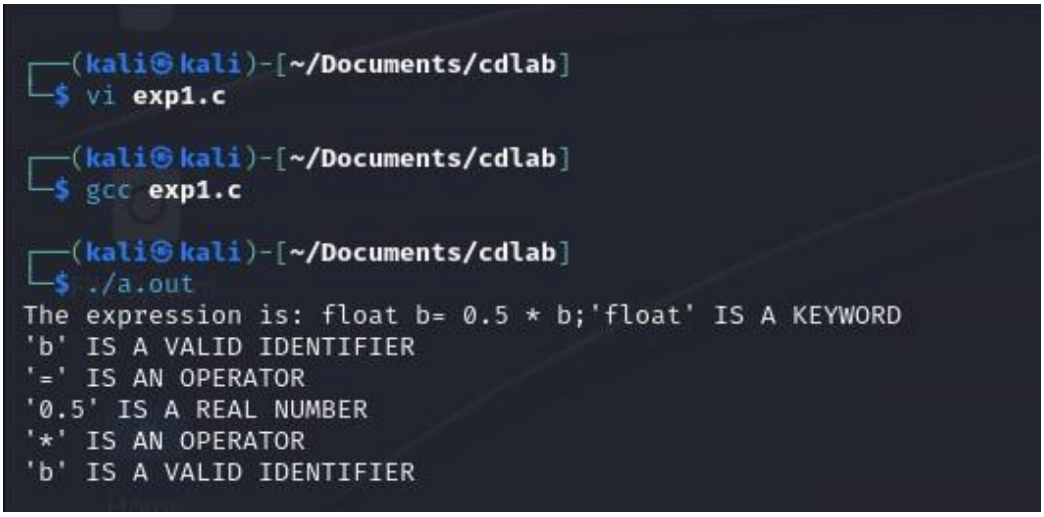
```
                    else if (validIdentifier(subStr) == true
                                && isDelimiter(str[right - 1]) == false)
                        printf("'%s' IS A VALID IDENTIFIER\n", subStr);

                    else if (validIdentifier(subStr) == false
                                && isDelimiter(str[right - 1]) == false)
                        printf("'%s' IS NOT A VALID IDENTIFIER\n", subStr);
                    left = right;}}
        return;}
int main(){
        // maximum length of string is 100 here
        printf("The expression is: float b= 0.5 * b;\n");
        char str[100] = "float b = 0.5 * b; ";

        parse(str); // calling the parse function

        return (0);
}
```

OUTPUT:



RESULT:

Thus, a C program is implemented to identify C keywords, identifiers, operators and end statements.

Exp No: 2

Date:

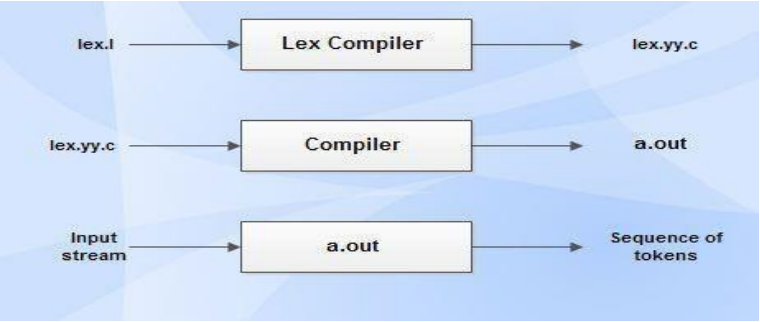
IMPLEMENT A LEXICAL ANALYZER TO COUNT THE NUMBER OF WORDS USING LEX TOOL

AIM:

To implement the program to count the number of words in a string using LEX tool.

STUDY:

Lex is a tool in lexical analysis phase to recognize tokens using regular expression. Lex tool itself is a lex compiler.



- lex.l is an a input file written in a language which describes the generation of lexical analyzer. The lex compiler transforms lex.l to a C program known as lex.yy.c.
- lex.yy.c is compiled by the C compiler to a file called a.out.
- The output of C compiler is the working lexical analyzer which takes stream of input characters and produces a stream of tokens.
- yylval is a global variable which is shared by lexical analyzer and parser to return the name and an attribute value of token.
- The attribute value can be numeric code, pointer to symbol table or nothing.
- Another tool for lexical analyzer generation is Flex.

STRUCTURE OF LEX PROGRAMS:

Lex program will be in following form declarations

%%

translation rules

%%

auxiliary functions

Roll Number: 210701089
Name: Jeciyazhini J

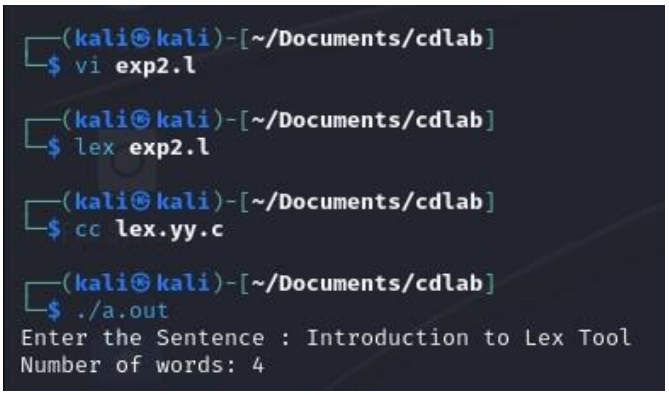
ALGORITHM:

1. Initialize counters for line count (lc), space count (sc), tab count (tc), character count (ch), and word count (wc).
2. Define rules to match newline, space, tab, and non-space/tab/newline characters. Increment corresponding counters based on matches.
3. Prompt the user to enter a sentence.
4. Invoke lexical analysis using yylex().
5. Signal the end of input.
6. Display the total word count.

PROGRAM:

```
%{
#include<stdio.h>
int lc=0,sc=0,tc=0,ch=0,wc=0;
%}
%%
[\n] { lc++; ch+=yyleng;}
[ \t] { sc++; ch+=yyleng;}
[^\t] { tc++; ch+=yyleng;}
[^\t\n ]+ { wc++; ch+=yyleng;}
%%
int yywrap(){ return 1; }
int main(){
        printf("Enter the Sentence : ");
        yylex();
        printf("Number of words: %d\n",wc);
        return 0;
}
```

OUTPUT:



RESULT:

Thus, the program to count the number of words in a string using LEX tool has been implemented.