EXP NO: 4

DESIGN AND IMPLEMENT A DESK CALCULATOR USING THE LEX TOOL

AIM:

To design and implement a Desk Calculator using the LEX tool**,** which validates arithmetic expressions containing +, -, *, /, numbers, and parentheses. The program ensures that the expression follows correct arithmetic syntax rules.

PROGRAM:

```
%{
#include <stdio.h>
int isValid = 1;
%}
%option noyywrap
%%
[0-9]+(\.[0-9]+)?      { printf("Number: %s\n", yytext); }
[\+\-\*/]              { printf("Operator: %s\n", yytext); }
"("                    { printf("Left Parenthesis: %s\n", yytext); }
")"                    { printf("Right Parenthesis: %s\n", yytext); }
[ \t]+                 ;
.                      { printf("Error: Invalid token '%s'\n", yytext); isValid = 0; }
%%
int main() {
    printf("Enter an arithmetic expression:\n");
    yylex();
    if (isValid)
        printf("Valid arithmetic expression.\n");
    else
        printf("Invalid arithmetic expression.\n");
    return 0;
}
```

OUTPUT :

```
keerthika@LAPTOP-SGM8D7IG:~$ vi exp_4.l
keerthika@LAPTOP-SGM8D7IG:~$ vi exp_4.l
keerthika@LAPTOP-SGM8D7IG:~$ lex exp_4.l
keerthika@LAPTOP-SGM8D7IG:~$ gcc lex.yy.c -o calculator
keerthika@LAPTOP-SGM8D7IG:~$ ./calculator
Enter an arithmetic expression:
7*8+9
Number: 7
Operator: *
Number: 8
Operator: +
Number: 9
```

RESULT:

Thus the above program reads an arithmetic expression, tokenizes it using **LEX rules**, and validates the syntax by recognizing **numbers, operators (+, -, *, /), and parentheses**. If the expression is **valid**, it prints "Valid arithmetic expression." Otherwise, it detects and reports **invalid tokens**

KEERTHIKA S 220701127