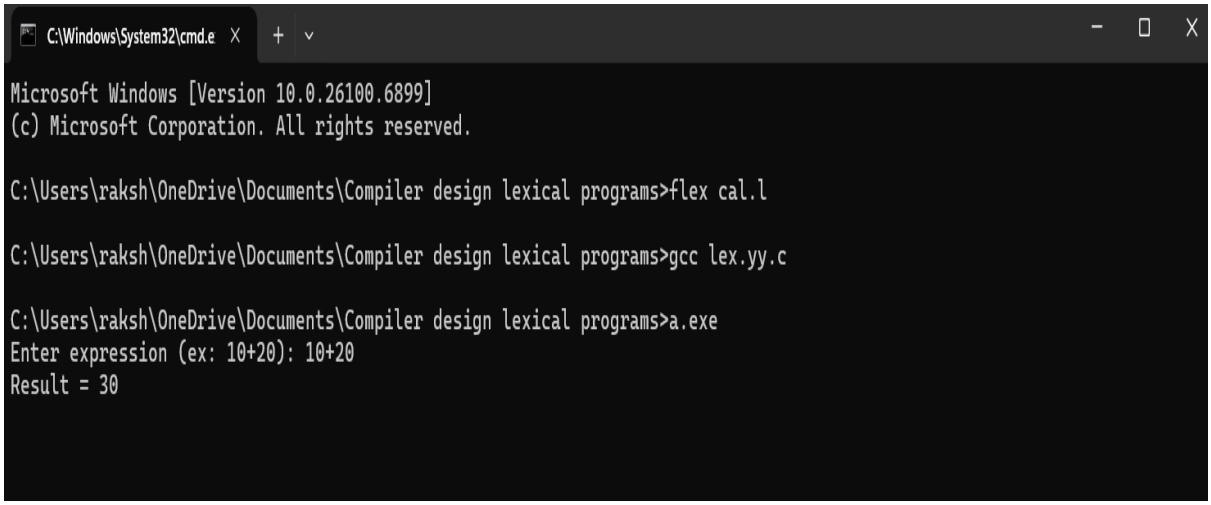
Experiment -40

Write a LEX program to implement basic mathematical operations.

Program:

```
%{
#include <stdio.h>
int a, b;
%}
%%
[0-9]+   { a = atoi(yytext); }
[\+\-\*\/]  {
            char op = yytext[0];
            yylex();
            b = atoi(yytext);
            switch(op)
            {
               case '+': printf("Result = %d\n", a + b); break;
               case '-': printf("Result = %d\n", a - b); break;
               case '*': printf("Result = %d\n", a * b); break;
               case '/': printf("Result = %d\n", a / b); break;
            }
         }
[ \t\n]+   ;
.          ;
%%
int yywrap() { return 1; }
int main()
{
   printf("Enter expression (ex: 10+20): ");
   yylex();
   return 0;}
```

Output:


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.
C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>flex cal.l
C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>gcc lex.yy.c
C:\Users\raksh\OneDrive\Documents\Compiler design lexical programs>a.exe
Enter expression (ex: 10+20): 10+20
Result = 30