PRACTICAL-8

AIM: Write a program to implement a calculator using lex and YACC

.Example program:

prct.y

```
%{
   #include<stdio.h>
   int flag=0;
%}
%token NUMBER
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'
%%

ArithmeticExpression: E{
        printf("\nResult=%d\n",$$);
        return 0;
       };
E:E'+'E {$$=$1+$3;}
 |E'-'E {$$=$1-$3;}
 |E'*'E {$$=$1*$3;}
 |E'/'E {$$=$1/$3;}
 |E'%'E {$$=$1%$3;}
 |'('E')' {$$=$2;}
 | NUMBER {$$=$1;}
;

%%
void main()
{
  printf("\nEnter Any Arithmetic Expression which can have operations Addition, Subtraction,
Multiplication, Divison, Modulus and Round brackets:\n");
  yyparse();

 if(flag==0)
  printf("\nEntered arithmetic expression is Valid\n\n");
```

```
}
void yyerror()

{
  printf("\nEntered arithmetic expression is Invalid\n\n");
  flag=1;
}
```

prct.l

```
%{
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}
%%
[0-9]+ {
        yylval=atoi(yytext);
        return NUMBER;
      }
[\t] ;
[\n] return 0;
. return yytext[0];
%%
int yywrap()
{
return 1;
}
```

Output:

```
[18012011033@linuxserv ~]$ vi prct8.l
[18012011033@linuxserv ~]$ vi prct8.y
[18012011033@linuxserv ~]$ yacc -d prct8.y
[18012011033@linuxserv ~]$ lex prct8.l
[18012011033@linuxserv ~]$ gcc lex.yy.c y.tab.c -w
[18012011033@linuxserv ~]$ ./a.out

Enter Any Arithmetic Expression which can have operations Addition, Subtraction,
 Multiplication, Divison, Modulus and Round brackets:
(5+5)*10

Result=100

Entered arithmetic expression is Valid

[18012011033@linuxserv ~]$
```