Practical : 7

Aim: Write a program to implement calculator using lex and YACC.

Code: **Lex file: p7.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;
}
```

YACC file: p7.c

```
%{
   #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;
}
```

Output:

```
Enter Any Arithmetic Expression which can have operations Addition, Subtraction, Multiplication, Divison, Modulus and Round brackets:
((5+10+5+6+4)/5)%2

Result=0

Entered arithmetic expression is Valid
```