# Practical No: 5

Name : Patel Savankumar P.
Enroll No : 19BCE519
Subject : Compiler Construction

**AIM: To implement a calculator in YACC..**

**Code:**

**File 1: Calc_Y.y**

```
%{
/* Definition section */
#include<stdio.h>
int flag=0;
%}

%token NUMBER

%left '+' '-'

%left '*' '/' '%'

%left '(' ')'

/* Rule Section */
%%

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;}

;

%%

//driver code
void main()
{
printf("\nEnter Any Arithmetic Expression having % / * - + ( ) :\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;
}
```

## File 2: Calc_L.l

```
%{
/* Definition section */
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}

/* Rule Section */
%%
[0-9]+ {
        yylval=atoi(yytext);
        return NUMBER;

    }
[\t] ;
" " ;

[\n] return 0;

. return yytext[0];

%%

int yywrap()
{
return 1;
}
```

## Execution Sequence:

```
PS E:\Semester 7\CC\Lab> bison -dy .\Calc_Y.y
PS E:\Semester 7\CC\Lab> flex .\Calc_L.l
PS E:\Semester 7\CC\Lab> gcc .\lex.yy.c .\y.tab.c -w
```

## Output:

```
PS E:\Semester 7\CC\Lab> .\a.exe

Enter Any Arithmetic Expression having / * - + ( ) :
5*10+5

Result=55

Entered arithmetic expression is Valid

PS E:\Semester 7\CC\Lab> .\a.exe

Enter Any Arithmetic Expression having / * - + ( ) :
5-2-5+7-6*8

Result=-43

Entered arithmetic expression is Valid
```

## Conclusion:

From this practical I learned how to create a calculator which can process a input expression using lexical and semantic rules using flex and bison.