Exercise – 8

1. Yacc Program to evaluate a given arithmetic expression

Examples:

Input: 7*(5-3)/2
Output: 7

Input: 6/((3-2)*(-5+2))
Output: -2

LEX PROGRAM:

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

%%
[0-9]+    {
            yylval = atoi(yytext);
            return NUMBER;
          }

[a-zA-Z]+    { return ID; }
[ \t]+        ;  /*For skipping whitespaces*/

\n            { return 0; }
.           { return yytext[0]; }

%%
```

YACC PROGRAM:

```
%{
   /* Definition section */
  #include <stdio.h>
%}

%token NUMBER ID
// setting the precedence
// and associativity of operators
%left '+' '-'
%left '*' '/'

/* Rule Section */
%%
E : T       {
              printf("Result = %d\n", $$);
              return 0;
            }

T :
   T '+' T { $$ = $1 + $3; }
   | T '-' T { $$ = $1 - $3; }
   | T '*' T { $$ = $1 * $3; }
   | T '/' T { $$ = $1 / $3; }
   | '-' NUMBER { $$ = -$2; }
   | '-' ID { $$ = -$2; }
   | '(' T ')' { $$ = $2; }
   | NUMBER { $$ = $1; }
   | ID { $$ = $1; };
% %

int main() {
   printf("Enter the expression\n");
   yyparse();
}

/* For printing error messages */
int yyerror(char* s) {
   printf("\nExpression is invalid\n");
}
```

2. YACC program to implement a Calculator and recognize a valid Arithmetic expression

Input: 4+5
Output: Result=9
Entered arithmetic expression is Valid

Input: 10-5

Output: Result=5
Entered arithmetic expression is Valid

Input: 10+5-
Output:
Entered arithmetic expression is Invalid

Input: 10/5
Output: Result=2
Entered arithmetic expression is Valid

Input: (2+5)*3
Output: Result=21
Entered arithmetic expression is Valid

Input: (2*4)+
Output:
Entered arithmetic expression is Invalid

Input: 2%5
Output: Result=2
Entered arithmetic expression is Valid

LEX PROGRAM

```
%{
/* 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;
```

```
}
```

YACC PROGRAM

```
%{
/* 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 which
                                    can have operations Addition,
                                    Subtraction, Multiplication, Division,
                                                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;
}
```