Exp No: 8 **Date**: 19/02/2025
Name: Y. Anusha **Regd No**: 22501A05J7

Experiment – 8

Aim: a. Implement a Yacc program to evaluate a given arithmetic expression.

Program:

Exp 8-1.l:

```
%{
   /* Definition section*/
   #include "y.tab.h"
   extern int yylval;
%}
%%
[0-9]+ {
   yylval = atoi(yytext);
   return NUMBER;
}
[a-zA-Z]+ {
    return ID;
}
[ \t]+ ; /*For skipping whitespaces*/
\n {
   return 0;
}
. {
   return yytext[0];
}
%%
```

Exp 8-1.y:

```
%{
   /* 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");
}
```

Output:

Case 1 :

```
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ lex exp8-1.l
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ byacc -d exp8-1.y
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ gcc lex.yy.c y.tab.c -ll
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ ./a.out
Enter the expression
6/((5-7)*(1+2))
Result = -1
(base) pllab@pllab:~/22501A05I3-CD/Exp8$
```

Case 2:

```
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ lex exp8-1.l
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ byacc -d exp8-1.y
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ gcc lex.yy.c y.tab.c -ll
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ ./a.out
Enter the expression
3*(9-7)/2
Result = 3
(base) pllab@pllab:~/22501A05I3-CD/Exp8$
```

Conclusion: Yacc program to evaluate a given arithmetic expression has been implemented successfully.

Exp No: 8 **Date**: 19/02/2025
Name: Y. Anusha **Regd No**: 22501A05J7

Aim: b. YACC program to implement a Calculator and recognize a valid Arithmetic expression

Program:

Exp 8-2.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;
}
```

Exp 8-2.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; }
;
%%
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;
}
```

Output:

Case 1:

```
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ lex exp8-2.l
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ byacc -d exp8-2.y
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ gcc lex.yy.c y.tab.c -ll
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ ./a.out

Enter Any Arithmetic Expression which can have operations Addition, Subtraction, Multiplication, Division, Modulus and Round brackets:
(8*7)/

Entered arithmetic expression is Invalid
```

Case 2:

```
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ lex exp8-2.l
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ byacc -d exp8-2.y
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ gcc lex.yy.c y.tab.c -ll
(base) pllab@pllab:~/22501A05I3-CD/Exp8$ ./a.out

Enter Any Arithmetic Expression which can have operations Addition, Subtraction, Multiplication, Division, Modulus and Round brackets:
7*(9/3)

Result=21

Entered arithmetic expression is Valid
```

Conclusion: YACC program to implement a Calculator and recognize a valid Arithmetic expression has been implemented successfully.