Date: 01/09/2025

Experiment No: 04

Aim: To write a program in YACC for parser generation.

Code:

```
%{
#include <stdio.h>
#include <stdlib.h>

int yylex(void);
int yyerror(const char *s);
%}

%union {
    double dval;
}

%token <dval> NUMBER
%left '+' '-'
%left '*' '/'
%right UMINUS

%type <dval> expr

%%

lines:
      lines expr '\n'   { printf("%g\n", $2); }
    | lines '\n'
    | /* empty */
    ;

expr:
      expr '+' expr     { $$ = $1 + $3; }
    | expr '-' expr     { $$ = $1 - $3; }
    | expr '*' expr     { $$ = $1 * $3; }
    | expr '/' expr     { $$ = $1 / $3; }
    | '(' expr ')'      { $$ = $2; }
    | '-' expr %prec UMINUS { $$ = -$2; }
    | NUMBER            { $$ = $1; }
    ;

%%

int yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
    return 0;
}
```

```c
#include <ctype.h>

int yylex(void) {
    int c;

    // skip spaces and tabs
    while ((c = getchar()) == ' ' || c == '\t')
        ;

    if (c == EOF)
        return 0;

    // detect numbers (integer or floating-point)
    if (c == '.' || isdigit(c)) {
        ungetc(c, stdin);
        double val;
        if (scanf("%lf", &val) == 1) {
            yylval.dval = val;
            return NUMBER;
        }
    }

    // return operator or newline as token
    return c;
}
```

Output:

```
asecomputerlab@hp-desktop:~/Desktop/22075$ bison -d calc.y
asecomputerlab@hp-desktop:~/Desktop/22075$ gcc calc.tab.c -o calc -lm
asecomputerlab@hp-desktop:~/Desktop/22075$ ./calc
1+2*3
7
(4+5)
9
(4+5)/2
4.5
```

Conclusion:

Thus, the program in YACC for parser generation has been executed successfully.