Date: 8/08/2025

Experiment No: 05

Aim: To implement Symbol Table.

Code:

```
%{
#include <stdio.h>
#include <ctype.h>
#define YYSTYPE double

int yylex();
int yyerror(const char *s);
%}
%token NUMBER
%left '+' '-'
%left '*' '/'
%right UMINUS
%%
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 %prec UMINUS { $$ = -$2; }
  | '(' expr ')' { $$ = $2; }
  | NUMBER
;
%%
int yylex() {
    int c;

    // Skip whitespace
    while ((c = getchar()) == ' ' || c == '\t');

    if (c == '.' || isdigit(c)) {
        ungetc(c, stdin);
        scanf("%lf", &yylval);
        return NUMBER;
    }    return c;
}
int yyerror(const char *s) {
```

```
}
int yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
    return 1;
}
int main() {
    return yyparse();
}
int yywrap() {
    return 1;
}
```

Output:

```
ubuntu:~$ yacc 4.y
ubuntu:~$ gcc -o 4 y.tab.c
ubuntu:~$ ./4
20+51
71
11+22
33
3463846+373623
3.83747e+06
323-121
202
3212+1616
4828
22074+22078
44152
```

Results: The program in YACC for parser generation has been executed successfully