**4. Implementation of Parser Using YACC and LEX**

LEX:

%{

#include<stdio.h>

#include<stdlib.h>

#include "y.tab.h"

%}

%%

[0-9]+(\.[0-9]+)?([eE][0-9]+)? {yylval.f=atof(yytext);return NUM;}

[+-/\*()] {return yytext[0];}

[\t\n\f\v] {;}

%%

int yywrap(void){

return 1;

}

YACC:

%{

#include<stdio.h>

#include<stdlib.h>

extern int yylex();

void yyerror(char \*msg);

%}

%union{

float f;

}

%token <f> NUM

%type <f> E T F

%%

S : E {printf("%f\n",$1);}

;

E : E '+' T {$$=$1+$3;} | E

'-' T {$$=$1-$3;} | T

{$$=$1;}

;

T : T '\*' F {$$=$1\*$3;} | T

'/' F {$$=$1/$3;} | F

{$$=$1;}

;

F : '(' E ')' {$$=$2;} |

'-' F {$$=$2;}

| NUM {$$=$1;}

;

%%

void yyerror(char \*msg){

fprintf(stderr,"%s\n",msg);

exit(0);

}

int main(void){

yyparse();

return 1;

}

**Output:**

C:\Users\16cseb47\Desktop\Ex-4>flex lex1.l

C:\Users\16cseb47\Desktop\Ex-4>bison -dy ya.y

C:\Users\16cseb47\Desktop\Ex-4>gcc lex.yy.c y.tab.c

C:\Users\16cseb47\Desktop\Ex-4>a.exe

3\*5+4

19.000000