Practical 5

Name: Tushar Harsora

Roll No.: 19BCE509

# Overview

There are 2 files for practical. First file reads the input stream and tokenizes it and gives to parse. The second file is yacc parser which checks for syntax and evaluates the expression.

# Code

practical.l

|  |
| --- |
| %{  #include <stdio.h>  #include "y.tab.h"  %}  %%  [0-9]+ |  [0-9]\*.[0-9]+ { yylval.value = atof(yytext); **return** NUMBER; }  ";" { **return** SEPERATOR; }  [ \n\t]+ { }  . { **return** yytext[0]; }  %% |

practical.y

|  |
| --- |
| %{  #define YYDEBUG 1  #include "y.tab.h"  #include <stdio.h>  int yylex();  void yyerror(char\*);  %}  %**union**{  float value;  }  %define parse.error detailed  *// %define parse.trace*  %left '+''-'  %left '\*''/'  %left MINUS  %token <value> NUMBER  %token SEPERATOR  %type <value> E  %type <value> S  %%  list : S |  list S  {    }  S : E SEPERATOR  {  $$ = $1;  printf("Result of Equation is %f**\n**", $$);  }  E : E '+' E  {  $$ = $1 + $3;  }  | E '-' E  {  $$ = $1 - $3;  }  | E '\*' E  {  $$ = $1 \* $3;  }  | E '/' E  {  $$ = $1 / $3;  }  | NUMBER  {  $$ = $1;  }  | '('E')'  {  $$ = $2;  }  | '-'E %prec MINUS  {  $$ = -$2;  }  %%  void yyerror(char\* s) {  printf("ERROR: %s**\n**", s);  }  int main(){  yyparse();  **return** 0;  } |

# Screenshots



