

**GB071 – Construção de Compiladores**  
**Tarefa – Gerador Automático de Analisadores Sintáticos (YACC)**

A partir dos códigos da versão mais completa da calculadora que adota analisadores léxico e sintático separados (discutidos na aula sobre YACC e disponibilizados abaixo), gere uma nova versão que, além do resultado, também apresenta a expressão digitada na notação pós-fixa. Por exemplo:

```
ubuntu@pc:/PosFix$ flex Lexer.lex
ubuntu@pc:/PosFix$ yacc -d Parser.y
ubuntu@pc:/PosFix$ gcc -o posfix Lexer.c y.tab.c
ubuntu@pc:/PosFix$ ./posfix
```

Digite a expressao desejada:

5.45\*(3.13+2)-(4/-2)

**5.45 3.13 2 + \* 4 2 (-) / - = 29.96**

5+(3-2)\*4

**5 3 2 - 4 \* + = 9.00**

4--2

**4 2 (-) - = 6.00**

*OBS: Digite <ENTER> na linha SEM expressão para encerrar a execução do programa*

## Código do Analisador Sintático (Parser.y)

```
%{
    #include "lexer.h" // Analisador lexico

    #define YYSTYPE double // Define tipo da pilha

    void yyerror(char *); // Prototipo da funcao de erro
}%

%token NUM

%left '+' '-'
%left '*' '/'
%right NEGAR

%%

lines : lines expr '\n' { printf("%.2lf\n", $2); }
      | lines '\n'      { return 0; }
      |
      | error '\n' { yyerror("Erro na ultima linha"); yyerrok; }
expr  : expr '+' expr { $$ = $1 + $3; }
      | expr '-' expr { $$ = $1 - $3; }
      | expr '*' expr { $$ = $1 * $3; }
      | expr '/' expr { $$ = $1 / $3; }
      | '(' expr ')'   { $$ = $2; }
      | '-' expr %prec NEGAR { $$ = -$2; }
      | NUM
      ;

%%

/* yacc error handler */
void yyerror(char * s) {
    fprintf (stderr, "%s\n", s);
}

int main(void) {
    return yyparse();
}
```

## Código do Analisador Léxico (*Lexer.lex*)

```
%option noyywrap

%option outfile="Lexer.c" header-file="Lexer.h"

% {

    #define YYSTYPE double    // Define tipo yylval (extern YYSTYPE yylval no y.tab.h)

    #include "y.tab.h"        // Biblioteca Analisador Sintatico

%}

DIGITO [0-9]

%%

[\t ]                { } /* ignora tabulacao e espaco */
{DIGITO}+([.]{DIGITO}+)? { yylval = atof (yytext); return NUM; }
.                    { return yytext[0]; }
\n                   { return yytext[0]; }

%%

// Seção de Código VAZIA
```