https://github.com/roxanazachman01/FLCD

Documentation:

Started work on yacc:

```
%{
#include <stdio.h>
#include <stdlib.h>
#define YYDEBUG 1

#define TIP_INT 1
#define TIP_REAL 2
#define TIP_CAR 3

double stack[20];
int sp;

void push(double x)
{ stack[sp++] = x; }

double pop()
{ return stack[--sp]; }

%}

%union {
        int l_val;
        char *p_val;
}
```

```
%token CHAR STRING INT IF ELSE READ WRITE WHILE BEGIN END
%token ' ' '(' ')' '[' ']' '{' '}' ';' '\n' '\t' '\r'
%token '+' '-' '*' '/' LE '<' GE '>' NE EQ '=' '%'

%token IDENTIFIER
%token <p_val> CONSTNR
%token <p_val> CONSTCHAR
%token <p_val> CONSTSTRING
%token EPSILON
%%

program: '{' decllist cmpdstmt '}'
                        ;

decllist: declaration decllistlala
;
decllistlala: decllist | EPSILON
;
declaration: type IDENTIFIER ';'
;
type: type1 arraydecl
;
type1: INT | CHAR | STRING
;
arraydecl: CONSTNR ']' | EPSILON
;
cmpdstmt: BEGIN stmtlist END
;
stmtlist: stmt stmtlistlala
;
stmtlistlala: stmtlist | EPSILON
```

;

stmt: simplstmt ';' | structstmt

;

simplstmt: assignstmt | iostmt

;

assignstmt: IDENTIFIER assignstmtlala '=' expression

;

assignstmtlala: '[' assignstmtlala2 ']' | EPSILON

;

assignstmtlala2: CONSTNR | IDENTIFIER

;

expression: term expression1

;

expression1: '+' expression1 | '-' expression1 | EPSILON | expression

;

term: factor term1

;

term1: '*' term1 | '%' term1 | '/' term1 | EPSILON

;

factor: '(' expression ')' | IDENTIFIER assignstmtlala | CONSTNR

;

iostmt: READ '(' IDENTIFIER assignstmtlala ')' | WRITE '(' writeitem ')'

;

writeitem: lalaitem | IDENTIFIER assignstmtlala

;

lalaitem: CONSTNR | CONSTCHAR | CONSTSTRING

;

structstmt: ifstmt | whilestmt

;

ifstmt: IF '(' condition ')' '{' stmtlist '}' elsestmt

;

```
elsestmt: ELSE '{' stmtlist '}' | EPSILON
;
whilestmt: WHILE '(' condition ')' '{' stmtlist '}'
;
condition: expression relation expression
;
relation: '<' | LE | GE | '>' | EQ | NE
;

%%

yyerror(char *s)
{
  printf("%s\n", s);
}

extern FILE *yyin;

main(int argc, char **argv)
{
  if(argc>1) yyin = fopen(argv[1], "r");
  if((argc>2)&&(!strcmp(argv[2],"-d"))) yydebug = 1;
  if(!yyparse()) fprintf(stderr,"\tO.K.\n");
}
```