Github: github.com/AntonioSuciu/FLCD

Flex flex.l

Bison -dy lang.y

gcc lex.yy.c y.tab.c

./a.out p1.txt

LANG.Y:

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

int yylex();
void yyerror();

%}

%token CHAR STRING INT IF ELSE READ WRITE WHILE
%token PRD PRI PPD PPI AD AI COLON
%token PLUS MINUS MULTIPLY DIV LE LT GE GT NE EQ ASSIGN MOD

%token IDENTIFIER
%token CONSTNR
%token CONSTCHAR
%token CONSTSTRING

%%
```

```
program: declaration_list statements
;
declaration_list: declaration declaration_list
                |
;
declaration: var_type IDENTIFIER eq_expr COLON
;
eq_expr: ASSIGN expr
                |
;
var_type: INT
                | CHAR
                | STRING
                ;
expr: term sign_and_expr
;
sign_and_expr: sign expr
                |
;
sign: PLUS
                | MINUS
                | MULTIPLY
                | DIV
                | MOD
;
term: IDENTIFIER
                | CONSTNR;
statements: statement statements
                |
```

```
;
statement: simple_stmt
                | struct_stmt
;
simple_stmt: assignment_stmt
                | input_output_stmt
;
struct_stmt: if_stmt
                | while_stmt
;
assignment_stmt: IDENTIFIER ASSIGN expr COLON
;
input_output_stmt: READ PRD term PRI COLON
                | WRITE PRD term PRI COLON
;
if_stmt: IF PRD condition PRI AD statements AI else_stmt
;
else_stmt: ELSE AD statements AI
                |
;
while_stmt: WHILE PRD condition PRI AD statements AI
;
condition: expr relation expr
;
relation: EQ
                | NE
                | LT
                | GT
                | LE
```

```
                        | GE
;



%%

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

extern FILE *yyin;

int 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");
}
```