<https://github.com/dahornea/FLCD/tree/main/Lab9>

%{

#define YY\_DECL int yylex()

extern YY\_DECL;

%}

%{

#include <stdio.h>

#include <stdlib.h>

int yyerror(char \*s);

#define YYDEBUG 1

%}

%token READ

%token DATA

%token WRITE

%token IF

%token ELSE

%token WHILE

%token STRUC

%token ENDIF

%token ENDWHILE

%token PLUS

%token MINUS

%token TIMES

%token DIV

%token LT

%token LE

%token EQ

%token NE

%token GE

%token GT

%token EQUAL

%token SQBRACKETOPEN

%token SQBRACKETCLOSE

%token COLON

%token OPEN

%token CLOSE

%token COMMA

%token DOT

%token IDENTIFIER

%token INTCONSTANT

%token STRINGCONSTANT

%start Program

%%

Program : Statements

;

Statements : Statements Statement DOT

| Statement DOT

;

Statement : DeclarationStatement

| AssignmentStatement

| IfStatement

| WhileStatement

| WriteStatement

| StrucStatement

| ReadStatement

;

DeclarationStatement : DATA COLON Identifiers

;

Identifiers : Identifiers COMMA IDENTIFIER

| IDENTIFIER

;

AssignmentStatement : IDENTIFIER EQUAL READ DOT

| IDENTIFIER EQUAL Expression DOT

| DATA COLON IDENTIFIER EQUAL IDENTIFIER DOT

;

Expression : Expression PLUS Term

| Expression MINUS Term

| Term

;

Term : Term TIMES Factor

| Term DIV Factor

| Factor

;

Factor : OPEN Expression CLOSE

| IDENTIFIER

| INTCONSTANT

| MINUS IDENTIFIER

;

StrucStatement : STRUC COLON IDENTIFIER SQBRACKETOPEN ExpressionList SQBRACKETCLOSE

| STRUC COLON IDENTIFIER SQBRACKETOPEN SQBRACKETCLOSE

;

ExpressionList : Expression COMMA ExpressionList

| Expression

;

IfStatement : IF Condition CompoundStatement ENDIF DOT

| IF Condition CompoundStatement ELSE CompoundStatement ENDIF DOT

;

WhileStatement : WHILE Condition CompoundStatement ENDWHILE DOT

;

WriteStatement : WRITE COLON STRINGCONSTANT DOT

| WRITE COLON IDENTIFIER DOT

;

ReadStatement : READ COLON STRINGCONSTANT DOT

| READ COLON IDENTIFIER DOT

;

Condition : Expression Relation Expression

;

Relation : LT

| LE

| EQ

| NE

| GE

| GT

;

CompoundStatement : OPEN Statements CLOSE

;

%%

int yyerror(char \*s) {

printf("Error: %s", s);

}

extern FILE \*yyin;

int main(int argc, char\*\* argv) {

if (argc > 1)

yyin = fopen(argv[1], "r");

if (!yyparse())

fprintf(stderr, "\tOK\n");

}