```
%{
	#include<stdio.h>
	#include<stdlib.h>
	#define YYDEBUG 1
%}
%error-verbose

%token MAIN
%token INT
%token STRING
%token ARRAY
%token READ
%token WRITE
%token IF
%token ELSE
%token WHILE
%token FALS
%token IDENTIFIER
%token CONSTANT
%token COMMA
%token SEMI_COLON
%token OPEN_CURLY_BRACKET
%token CLOSED_CURLY_BRACKET
%token OPEN_RIGHT_BRACKET
%token CLOSED_RIGHT_BRACKET
%token OPEN_ROUND_BRACKET
%token CLOSED_ROUND_BRACKET
%token PLUS
%token MINUS
%token DIV
%token MUL
%token PERCENT
%token EQ

%start program

%%

program: MAIN OPEN_CURLY_BRACKET statement_list CLOSED_CURLY_BRACKET;
declaration: type identifier_list SEMI_COLON;
identifier_list: IDENTIFIER | IDENTIFIER COMMA identifier_list;
type: INT|STRING|ARRAY;
statement_list: statement SEMI_COLON | statement SEMI_COLON statement_list;
statement: declaration|simple_stmt|struct_stmt;
simple_stmt: assignment_stmt|io_stmt|return_stmt;
struct_stmt: compound_stmt|if_stmt|while_stmt;
assignment_stmt: IDENTIFIER EQ expression;
return_stmt: RAPORTEAZA IDENTIFIER|RAPORTEAZA CONSTANT;
io_stmt: read_stmt|print_stmt;
read_stmt: CITESTE OPEN_ROUND_BRACKET STD_IN COMMA IDENTIFIER
CLOSED_ROUND_BRACKET;
print_stmt: CITESTE OPEN_ROUND_BRACKET STD_OUT COMMA IDENTIFIER
CLOSED_ROUND_BRACKET;
compound_stmt: OPEN_CURLY_BRACKET statement_list CLOSED_CURLY_BRACKET;
if_stmt: simple_if|simple_if else|simple_if else_if else;
simple_if: DACA OPEN_ROUND_BRACKET condition CLOSED_ROUND_BRACKET
compound_stmt;
```

```
else: EVENTUAL compound_stmt;
else_if: DACA_NU OPEN_ROUND_BRACKET condition CLOSED_ROUND_BRACKET
compound_stmt;
while_stmt: IN_TIMP_CE OPEN_ROUND_BRACKET condition CLOSED_ROUND_BRACKET
compound_stmt;
condition: expression RELATION expression;
expression: term|expression operator expression
term: IDENTIFIER|CONSTANT|IDENTIFIER OPEN_SQUARE_BRACKET INTREG
CLOSED_SQUARE_BRACKET;
operator: PLUS|MINUS|DIV|MUL|POWER|PERCENT;


%%


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,"\t CORRECT\n");
}
```