

```

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

extern int yylineno;
extern char* yytext;
int yylex();
void yyerror(const char *s);
%}

%token INT
%token CHAR
%token LIST
%token WHILE
%token IF
%token ELSE
%token READ
%token WRITE

%token IDENTIFIER
%token INTEGER
%token STRING
%token CHARACTER

%token EQ
%token LE
%token GE
%token NOTEQ

%left '+' '-'
%left '/' '%' '*'
%left OR
%left AND

%%

program: compound_statement
        | program compound_statement
compound_statement: '{' statement_list '}'
statement_list: statement
               | statement statement_list
               ;
statement: declaration ';'
         | simple_statement ';'

```

```

    |   structured_statement
;
declaration:   type IDENTIFIER
              |   declaration_and_assignment
              |   array_declaration
;
declaration_list: declaration
                  | declaration declaration_list
;
type:   INT
      | CHAR
      | STRING
;
declaration_and_assignment: type IDENTIFIER '=' constant
;
constant:   INTEGER
          | CHARACTER
          | STRING
;
array_declaration: type IDENTIFIER '=' LIST '(' ')'
;
integer_constant_or_identifier: constant
                                | IDENTIFIER
;
simple_statement: assign_statement
                 | io_statement
;
assign_statement: IDENTIFIER '=' expression
                 | array_element '=' expression
expression: expression '+' term
           | expression '-' term
           | term
;
term: term '*' factor
     | term '/' factor
     | term '%' factor
     | factor
;
factor: '(' expression ')'
       | integer_constant_or_identifier
       | array_element
;
io_statement: READ '(' IDENTIFIER ')'
             | READ '(' array_element ')'
             | WRITE '(' IDENTIFIER ')'

```

```

    ;
array_element: IDENTIFIER '[' integer_constant_or_identifier ']'
    ;
structured_statement: | if_statement
    | while_statement
    ;
if_statement: IF '(' condition ')' '{' statement_list '}'
    | IF '(' condition ')' '{' statement_list '}' ELSE '{' statement_list '}'
    | IF '(' condition ')' '{' statement_list '}' ELSE if_statement
    ;
while_statement: WHILE '(' condition ')' '{' statement_list '}'
    ;
condition: expression relation condition
    | expression relation expression
    ;
relation: '<'
    | '>'
    | EQ
    | LE
    | GE
    | AND
    | OR
    | NOTEQ
    ;

```

```
%%
```

```

void yyerror(const char *s)
{
    printf("%s on line: %d for token: %s\n", s, yylineno, yytext);
}

```

```
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, "Syntactically correct.\n");
    return 0;
}

```