

```

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

%}

%token var
%token '='
%token Read
%token Write
%token if
%token end
%token ':'
%token not
%token and
%token or
%token xor
%token while
%token '+'
%token '-'
%token '*'
%token '/'
%token '%'
%token '<'
%token '>'
%token '=='
%token '!='
%token '>='
%token '<='
%%
declaration_statement : var identifier
                      | var identifier '=' expression
assignment_statement : identifier '=' expression
expression : expression operation term
           | term
term : const
     | identifier
input : Read >> identifier
output : Write << expression
        | Write << identifier
if_statement : if condition ':' statementlist end
             | if condition ':' statementlist else ':' statementlist end
condition : expression relation expression
          | condition logicalop condition
          | not condition
logicalop : and
          | or
          | xor
statement : simplestatement

```

```

    | struct_statement
simplestatement : assignment_statement
                | input
                | output
statementlist : statement
              | statement statementlist
struct_statement : if_statement
                 | while_statement
operation : +
          | -
          | *
          | /
          | %
relation : >
          | <
          | ==
          | !=
          | <=
          | >=
while_statement : while condition : statementlist

```

```
%%
```

```

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, "\t0.K.\n");
}

```