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

#define TIP_INT 1
#define TIP_REAL 2
#define TIP_CAR 3

double stiva[20];
int sp;

void push(double x)
{ stiva[sp++]=x; }

double pop()
{ return stiva[--sp]; }

%}

%union {
        int l_val;
        char *p_val;
}

%token local
%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 <=
%token Boolean
%token Char
%token Int
%token Float
%token String
%%
```

```
declstmt : local identifier
         | local identifier ':' type
         | local identifier ':' type = expression
assignstmt : identifier '=' expression
expression : expression operation term
         | term
term : const
     | identifier
input : read identifier
output : write expression
      | write identifier
ifstmt : if condition ':' stmtlist end
     | if condition ':' stmtlist else ':' stmtlist end
condition : expression relation expression
         | condition logicalop condition
         | not condition
logicalop : and
         | or
         | xor
stmt : simplestmt
     | structstmt
simplestmt : assignstmt ';'
         | input ';'
         | output ';'
stmtlist : stmt
         | stmt stmtlist
structstmt : ifstmt
             | whilestmt
operation : +
         | -
         | *
         | /
         | %
relation : >
        | <
         | ==
         | !=
         | <=
         | >=
whilestmt : while condition : stmtlist end
primitive : Boolean
         | Char
         | Int
         | Float
nonprimitive : String
type : primitive
     | nonprimitive

%%

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

```c
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,"\tO.K.\n");
}
```