```
Programma :  HEAD Decls START Statements

Decls : Var_decl Decls
                | Def_decl Decls
                | ε

Statements : Stat Statements
                         | ε

Var_decl : Type Var_decls_init SEMI

Type : INT  |  BOOL   |  DOUBLE   | STRING | CHAR

Var_decls_init : ID Var_init_value COMMA Var_decls_init
        | ID Var_init_value

Var_init_value : ASSIGN Expr
                | ε

Vars    → ID COMMA Vars
        | ID

Def_decl : DEF ID LPAR Par_decls RPAR Body
Def_decl : DEF ID LPAR  RPAR Body

Var_decls : Var_decl Var_decls
          | ε

Par_decls  : Par_type Type ID  COMMA Par_decls
                          | Par_type Type ID

Par_type : IN | OUT | INOUT

Body : LGPAR Var_decls Statements RGPAR

Stat :   Vars READ SEMI
        | Args WRITE SEMI
        | ID ASSIGN Expr SEMI
        | ID LPAR Args RPAR SEMI
        | ID LPAR RPAR SEMI
        | IF LPAR Expr RPAR THEN Comp_stat ELSE Comp_stat
        | IF LPAR Expr RPAR THEN Comp_stat
        | WHILE LPAR Expr RPAR DO Comp_stat

Vars : ID COMMA Vars
         | ID

Args :  Expr COMMA  Args
                | Expr

Comp_stat : LGPAR Statements RGPAR

Expr   : Expr Arith_op Expr
                | Expr Bool_op Expr
```

```
        | Expr Rel_op Expr
        | MINUS Expr
        | NOT Expr
        | LPAR Expr RPAR
        | TRUE
        | FALSE
        | ID
        | INT_CONST
        | DOUBLE_CONST
        | CHAR_CONST
        | STRING_CONST


Arith_op : PLUS  |  MINUS  | TIMES  |  DIV

Bool_op : AND | OR

Rel_op  :  GT | GE | LT | LE | EQ



Specifica lessicale

HEAD             head
START            start
SEMI            ;
INT             int
BOOL              bool
DOUBLE              double
STRING              string
CHAR              char
ID             jletter (jletter | jdigit)*
COMMA             ,
DEF             def
LPAR             (
RPAR             )
LGPAR             {
RGPAR            }
READ            <-
WRITE            ->
PLUS              +
MINUS             -
TIMES              *
DIV            /
INT_CONST            pattern per interi
DOUBLE_CONST       pattern per double
STRING_CONST        pattern per stringhe
CHAR_CONST
TRUE             true
FALSE            false
ASSIGN                =
IF             if
THEN                    then
WHILE            while
```

```
DO            do
ELSE            else
GT             >
GE             >=
LT              <
LE               <=
EQ               ==
NOT              not
AND              and
OR         or
UMINUS     -
IN               in
OUT              out
INOUT            inout
```