start -> program

program -> function
program -> epsilon

function -> FUNCTION IDENT SEMICOLON parameters declarations parameters declarations parameters statements parameters

parameters -> BEGIN_PARAMS
parameters -> END_PARAMS BEGIN_LOCALS
parameters -> END_LOCALS BEGIN_BODY
parameters -> END_BODY

declarations -> declaration SEMICOLON declarations
declarations -> epsilon

declaration -> identify COLON declaration_2

identify -> IDENT
identify -> IDENT COMMA identify

declaration_2 -> INTEGER
declaration_2 -> ARRAY L_SQUARE_BRACKET NUMBER R_SQUARE_BRACKET OF INTEGER

statements -> statement SEMICOLON statements
statements -> epsilon

statement -> var ASSIGN AS_expr
statement -> IF OR_expr THEN statements ENDIF
statement -> IF OR_expr THEN statements ELSE statements ENDIF
statement -> WHILE OR_expr BEGINLOOP statements ENDLOOP
statement -> DO BEGINLOOP statements ENDLOOP WHILE OR_expr
statement -> FOR var ASSIGN NUMBER SEMICOLON OR_expr SEMICOLON var Assign AS_expr BEGINLOOP statements ENDLOOP
statement -> READ vars
statement -> WRITE vars
statement -> CONTINUE
statement -> RETURN AS_expr

OR_expr -> OR_expr OR AND_expr
OR_expr -> AND_expr

AND_expr -> NOT_expr AND AND_expr
AND_expr -> NOT_expr

NOT_expr -> NOT REL_expr
NOT_expr -> REL_expr

REL_expr -> AS_expr comp AS_expr
REL_expr -> L_PAREN OR_expr R_PAREN
REL_expr -> TRUE
REL_expr -> FALSE

comp -> GT
comp -> LT
comp -> GTE
comp -> LTE
comp -> EQ
comp -> NEQ

AS_expr -> MDM_expr AS_expr2

AS_expr2 -> epsilon
AS_expr2 -> SUB AS_exprAS_expr2 -> ADD AS_expr

MDM_expr -> NEG_term MDM_expr2

MDM_expr2 -> epsilon
MDM_expr2 -> MOD MDM_expr
MDM_expr2 -> MULT MDM_expr
MDM_expr2 -> DIV MDM_expr

NEG_term -> SUB term
NEG_term -> term
NEG_term -> IDENT term_id

term -> var
term -> var L_SQUARE_BRACKET AS_expr R_SQUARE_BRACKET
term -> L_PAREN AS_expr R_PAREN
term -> NUMBER

term_id -> L_PAREN term_ex R_PAREN
term_id -> L_PAREN R_PAREN

term_ex -> AS_expr COMMA term_ex

term_ex -> AS_expr

term_exp -> COMMA term_ex
term_exp -> epsilon

vars -> var COMMA vars
vars -> var

var -> IDENT
var -> IDENT L_SQUARE_BRACKET AS_expr R_SQUARE_BRACKET