program ::= | function program

function ::= FUNCTION identifier SEMICOLON BEGIN_PARAMS decline END_PARAMS BEGIN_LOCALS decline END_LOCALS BEGIN_BODY statline END_BODY

declaration ::= identifier COMMA declaration | identifier COLON INTEGER | identifier COLON ARRAY L_SQUARE_BRACKET number R_SQUARE_BRACKET OF INTEGER

number ::= NUMBER

decline ::= | declaration SEMICOLON decline

identifier ::= IDENT

statement ::= var ASSIGN expression | IF boolexp THEN statline stathelp ENDIF| WHILE boolexp BEGINLOOP statline ENDLOOP | DO BEGINLOOP statline ENDLOOP WHILE boolexp | FOR var ASSIGN number SEMICOLON boolexp SEMICOLON var ASSIGN expression BEGINLOOP statline ENDLOOP | READ varline| WRITE varline| CONTINUE | RETURN expression

stathelp ::= | ELSE statline

varline ::= var | var COMMA varline

statline ::= | statement SEMICOLON statline

boolexp ::= relationandexpr| relationandexpr OR boolexp

relationandexpr ::= relationexpr | relationexpr AND relationandexpr

relationexpr ::= NOT relationhelper | relationhelper

relationhelper ::= expression comp expression | TRUE | FALSE | L_PAREN boolexp R_PAREN

comp ::= EQ | NEQ | LT | GT | LTE | GTE

expression ::= multiplicativeexp | multiplicativeexp ADD expression | multiplicativeexp SUB expression

multiplicativeexp ::= term | term MULT multiplicativeexp | term DIV multiplicativeexp | term MOD multiplicativeexp

term ::= term1 | SUB term1 | identifier L_PAREN expresscomm R_PAREN

term1 ::= var | number | L_PAREN expression R_PAREN

var ::= identifier | identifier L_SQUARE_BRACKET expression R_SQUARE_BRACKET

expresscomm ::= expression | expression COMMA expresscomm