Based off of outputs on mytest.parse and the syntax diagram
Syntax diagram: https://www.cs.ucr.edu/~amazl001/teaching/cs152/S19/webpages2/syntax.html
mytest.parse: https://www.cs.ucr.edu/~amazl001/teaching/cs152/S19/webpages2/mytest.parse

prog_start: prog_start → functions

functions: functions → function functions | ε

function: function → FUNCTION ident SEMICOLON BEGIN_PARAMS declarations END_PARAMS BEGIN_LOCALS declarations END_LOCALS BEGIN_BODY statements END_BODY

declaration: declaration → identifiers COLON INTEGER | identifiers COLON ARRAY L_SQUARE_BRACKET OF INTEGER

declarations: declarations → declaration SEMICOLON declarations | ε

statements: statements → statement SEMICOLON statements | statement SEMICOLON

statement: statement → var ASSIGN | IF bool_expr THEN statements else_statement ENDIF | WHILE bool_expr BEGINLOOP statements ENDLOOP | DO BEGINLOOP statements ENDLOOP WHILE bool_expr | FOREACH ident IN ident BEGINLOOP statements ENDLOOP | READ vars | WRITE vars | CONTINUE | RETURN expression

else_statement: else_statement → ELSE statements | ε

bool_expr: bool_expr → relation_and_expr | relation_and_expr OR bool_expr

relation_and_expr: relation_and_expr → relation_expr | relation_expr AND relation_and_expr

relation_expr: relation_expr → NOT relation_expr | relation_expr

relation_exp: relation_exp → expression comp expression | TRUE | FALSE | L_PAREN bool_expr R_PAREN

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

expression: expression → multiplicative_expr | multiplicative_expr ADD expression | multiplicative_expr SUB expression

expressions: expressions → expression COMMA expressions | expression | ε

multiplicative_expr: multiplicative_expr → term MULT multiplicative_expr | term DIV multiplicative_expr | term MOD multiplicative_expr

var: var → ident L_SQUARE_BRACKET expression R_SQUARE_BRACKET | ident

vars: vars → var | var COMMA vars

term: term → var | SUB var | NUMBER | SUB NUMBER | L_PAREN expression R_PAREN | SUB L_PAREN expression R_PAREN | ident L_PAREN expressions R_PAREN

identifiers: identifiers → ident | ident COMMA identifiers

ident: ident → IDENT