Grammar02

terminal states: ALL CAPS
non-terminal states: lower case

program -> functions

functions -> function
| functions function

function -> FUNCTION IDENTIFIER SEMICOLON BEGINPARAMS declarations ENDPARAMS BEGINLOCALS declarations ENDLOCALS BEGINBODY statements ENDBODY

declarations -> EMPTY
| declarations declaration

declaration -> identifiers COLON INTEGER SEMICOLON
| identifiers COLON ARRAY L_SQUARE_BRACKET NUMBER R_SQUARE_BRACKET OF INTEGER SEMICOLON
| identifiers COLON ENUM L_PAREN enum_list R_PAREN SEMICOLON

identifiers -> IDENTIFIER
| identifiers COMMA IDENTIFIER

enum_list -> IDENTIFIER
| enum_list COMMA IDENTIFIER

statements -> EMPTY
| statements statement

statement -> var ASSIGN expression SEMICOLON
| IF bool_expr THEN statements ENDIF opt_semi
| IF bool_expr THEN statements ELSE statements ENDIF opt_semi
| WHILE bool_expr BEGINLOOP statements ENDLOOP opt_semi
| DO BEGINLOOP statements ENDLOOP WHILE bool_expr SEMICOLON
| FOREACH IDENTIFIER IN IDENTIFIER BEGINLOOP statements ENDLOOP opt_semi
| READ vars SEMICOLON
| WRITE vars SEMICOLON
| CONTINUE SEMICOLON
| RETURN expression SEMICOLON

```
opt_semi -> EMPTY
| SEMICOLON

vars -> var
| var COMMA vars

bool_expr -> relation_and_expr
| bool_expr OR relation_and_expr

relation_and_expr -> relation_expr
| relation_and_expr AND relation_expr

relation_expr -> expression comp expression
| NOT expression comp expression
| NOT L_PAREN bool_expr R_PAREN
| L_PAREN bool_expr R_PAREN
| expression
| TRUE
| FALSE

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

expression -> multiplicative_expr
| expression PLUS multiplicative_expr
| expression MINUS multiplicative_expr

multiplicative_expr -> term
| multiplicative_expr MULT term
| multiplicative_expr DIV term
| multiplicative_expr MOD term

term -> var
| NUMBER
| L_PAREN expression R_PAREN
| MINUS term
| IDENTIFIER L_PAREN expressions R_PAREN
```

var -> IDENTIFIER
| IDENTIFIER L_SQUARE_BRACKET expression R_SQUARE_BRACKET

expressions -> EMPTY
| expression
| expressions COMMA expression