Prog_start -> functions

Functions -> function function | (epsilon)

Function -> functions

Function -> FUNCTION IDENT SEMICOLON BEGIN_PARAMS declarations END_PARAMS BEGIN_LOCALS declarations END_LOCALS BEGIN_BODY statements END_BODY

Declarations -> declaration SEMICOLON declarations | (epsilon)

Declaration -> identifiers COLON INTEGER | identifiers COLON ENUM L_PAREN identifiers R_PAREN | identifiers COLON ARRAY L_SQUARE_BRACKET NUMBER R_SQUARE_BRACKET OF INTEGER

Identifiers -> IDENT | IDENT COMMA identifiers

Statement -> statement SEMICOLON | statement SEMICOLON statements

Statement -> var ASSIGN expression | IF bool_expression THEN statements ENDIF | IF bool_expression THEN statements ELSE statements ENDIF | WHILE bool_expression BEGINLOOP statements ENDLOOP | DO BEGINLOOP statements ENDLOOP WHILE bool_expression | FOR variables ASSIGN NUMBER SEMICOLON bool_expression SEMICOLON variables ASSIGN expression BEGINLOOP statements ENDLOOP | READ variables | WRITE variables | CONTINUE | RETURN expression

Bool_expression -> relation_and_expr | bool_expression OR relation_and_expr

Relation_and_expr -> relation_expr | relation_and_expr AND relation_expr

Relation_expr -> expression comp expression | NOT expression comp expression | TRUE | NOT TRUE | FALSE | NOT FALSE | L_PAREN bool_expression R_PAREN | NOT L_PAREN bool_expression R_PAREN

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

Expression-> multiplicative_expr | expression ADD multiplicative_expr | expression SUB multiplicative_expr

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

Term -> var | SUB var | NUMBER | SUB NUMBER| L_PAREN expression R_PAREN | IDENT L_PAREN expression R_PAREN | IDENT L_PAREN R_PAREN

```
Var -> IDENT | IDENT L_SQUARE_BRACKET expression R_SQUARE_BRACKET

Variables -> var COMMA variables | var

Ident -> IDENT

Number-> NUMBER
```