program -> **program id (** *identifier_list* **) ;** *program'*
program' -> *declarations program"*
program' -> *subprogram_declarations compound_statement* **.**
program' -> *compound_statement* **.**
program" -> *subprogram_declarations compound_statement* **.**
program" -> *compound_statement* **.**

identifier_list -> **id** *identifier_list'*
identifier_list' -> **,** **id** *identifier_list'*
identifier_list' -> **e**

declarations -> **var id :** *type* **;** *declarations'*
declarations' -> **var id :** *type* **;** *declarations'*
declarations' -> **e**

type -> *standard_type*
type -> **array [num .. num] of** *standard_type*

standard_type -> **integer**
standard_type -> **real**

subprogram_declarations -> *subprogram_declaration* **;** *subprogram_declarations'*
subprogram_declarations' -> *subprogram_declaration* **;** *subprogram_declarations'*
subprogram_declarations' -> **e**

subprogram_declaration -> *subprogram_head subprogram_declaration*'
subprogram_declaration' -> *declarations subprogram_declaration"*
subprogram_declaration' -> *compound_statement*
subprogram_declaration' -> *subprogram_declarations compound_statement*
subprogram_declaration" -> *compound_statement*
subprogram_declaration" -> *subprogram_declarations compound_statement*

subprogram_head -> **procedure id** *subprogram_head'*
subprogram_head' -> *arguments* **;**
subprogram_head' -> **;**

arguments -> **(** *parameter_list* **)**

parameter_list -> **id :** *type parameter_list'*
parameter_list' -> **; id :** *type parameter_list'*
parameter_list' -> **e**

compound_statement -> **begin** *compound_statement'*
compound_statement' -> *optional_statements* **end**
compound_statement' -> **end**

optional_statements -> *statement_list*

statement_list -> *statement statement_list'*

statement_list' -> **;** *statement statement_list'*
statement_list' -> **e**

statement -> *variable* **assignop** *expression*
statement -> *procedure_statement*
statement -> *compound_statement*
statement -> **if** *expression* **then** *statement statement'*
statement -> **while** *expression* **do** *statement_list*
statement' -> **else** *statement*
statement' -> **e**

variable -> **id** *variable'*
variable' -> **[** *expression* **]**
variable' -> **e**

procedure_statement -> **call id** *procedure_statement'*
procedure_statement' -> **(** *expression_list* **)**
procedure_statement' -> **e**

expression_list -> *expression expression_list'*
expression_list' -> **,** *expression expression_list'*
expression_list' -> **e**

expression -> *simple_expression expression'*
expression' -> **relop** *simple_expression*
expression' -> **e**

simple_expression -> *term simple_expression'*
simple_expression -> *sign term simple_expression'*
simple_expression' -> **addop** *term simple_expression'*
simple_expression' -> **e**

term -> *factor term'*
term' -> **mulop** *factor term'*
term' -> **e**

factor -> **num**
factor -> **(** *expression* **)**
factor -> **id** *factor'*
factor -> **not** *factor*
factor' -> **[** *expression* **]**
factor' -> **e**

sign -> **+**
sign -> **-**