Production Rules

program ->	*function_declarations*
	main()
	compound_statement
	function_definitions
identifierList ->	**id** |
	id , *identifier_list*
declarations ->	*type identifier_list* **;** *declarations* |
	ϵ
type ->	**void** |
	int |
	float
function_declarations ->	*function_declaration* **;** *function_declarations* |
	ϵ
function_declaration ->	*type id parameters*
function_definitions ->	*function_definition function_definitions* |
	ϵ
function_definition ->	*type id parameters compound_statement*
parameters ->	**(** *parameter_list* **)**
parameter_list ->	*type* **id** |
	type **id ,** *parameter_list*
compound_statement ->	**{** *declarations optional_statements* **}**
optional_statements ->	*statement_list* |
	ϵ

statement_list -> *statement* |
statement **;** *statement_list*

statement -> *variable* **assignop** *expression* |
procedure_statement |
compound_statement |
if *expression* **then** *statement* **else** *statement* |
while *expression* **do** *statement* |
read (id) |
write (*expression* **)** |
return *expression*

variable -> **id** |
id [*expression* **]**

procedure_statement -> **id** |
id (*expression_list* **)**

expression_list -> *expression* |
expression **,** *expression_list*

expression -> *simple_expression* |
simple_expression **relop** *simple_expression*

simple_expression -> *term simple_part* |
sign term simple_part

simple_part -> **addop** *term simple_part* | ϵ

term -> *factor term_part*

term_part -> **mulop** *factor term_part* | ϵ

factor -> **id** |
id [*expression* **]** |
id (*expression_list* **)** |
num |
(*expression* **)** |
! *factor*

sign -> **+** |
-

Lexical Conventions

1. Comments are surrounded by **/*** and ***/**. Alternately anything from **//** to the end of a line. Comments may appear after any token.

2. Blanks between tokens are optional.

3. Token **id** for identifiers matches a letter followed by letter or digits:
   letter -> [a-zA-Z]
   digit -> [0-9]
   id -> letter (letter | digit)*

The ***** indicates that the choice in the parentheses may be made as many times as you wish.

1. Token **num** matches numbers as follows:
   digits -> digit digit*
   optional_fraction -> . digits | λ
   optional_exponent -> (E (+ | - | λ) digits) | λ
   num -> digits optional_fraction optional_exponent

2. Keywords are reserved.

3. The relational operators (**relop**'s) are:
   ==, **!=**, **<**, **<=**, **>=**, and **>**.

4. The **addop**'s are **+**, **-**, and **||**.

5. The **mulop**'s are *****, **/**, **%**, and **&&**.

6. The lexeme for token **assignop** is **=**.