

Mini PASCAL Grammar

program →

program *id* (*identifier_list*) ;
declarations
subprogram_declarations
compound_statement
.

identifier_list →

id
| *identifier_list* , **id**

declarations →

declarations **var** *identifier_list* : *type* ;
| ϵ

type →

standard_type
| **array** [**num** .. **num**] **of** *standard_type*

standard_type →

integer
| **real**

subprogram_declarations →

subprogram_declarations *subprogram_declaration* ;
| ϵ

subprogram_declaration →

subprogram_head *declarations* *compound_statement*

subprogram_head →

function **id** *arguments* : *standard_type* ;
| **procedure** **id** *arguments* ;

arguments →

(*parameter_list*)
| ϵ

parameter_list →

identifier_list : *type*
| *parameter_list* ; *identifier_list* : *type*

compound_statement →

begin
optional_statements
end

optional_statements →
 statement_list
 | ε

statement_list →
 statement
 | *statement_list* ; *statement*

statement →
 variable **assignop** *expression*
 | *procedure_statement*
 | *compound_statement*
 | **if** *expression* **then** *statement* **else** *statement*
 | **while** *expression* **do** *statement*

variable →
 id
 | **id** [*expression*]

procedure_statement →
 id
 | **id** (*expression_list*)

expression_list →
 expression
 | *expression_list* , *expression*

expression →
 simple_expression
 | *simple_expression* **relop** *simple_expression*

simple_expression →
 term
 | *sign* *term*
 | *simple_expression* **addop** *term*

term →
 factor
 | *term* **mulop** *factor*

factor →
 id
 | **id** (*expression_list*)
 | **num**
 | (*expression*)
 | **not** *factor*

sign →
 + | -