The MINI-L Source Code Language

MINI-L Language Description

The MINI-L language provides the following program constructs.

1. Integer scalar variables.
2. One-dimensional&Two-domensional arrays of integers.
3. Assignment statements.
4. While and Do-While loops.
5. Continue statement.
6. If-then-else statements.
7. Read and write statements.
8. Comments.
9. Functions.

Here are some additional details of the MINI-L language.

- A *comment* is introduced by "##" and extends to the end of the current line.
- MINI-L is *case sensitive*. All reserved words are expressed in lower case.
- A valid *identifier* must begin with a letter, may be followed by additional letters, digits, or underscores, and cannot end in an underscore.
- Whitespace in MINI-L programs can occur due to regular blank spaces, tabs, or newlines.

Functions in MINI-L take some constant number of scalar arguments, and return a single scalar result. All arguments are passed by value (there are no reference arguments). A syntactically and semantically valid MINI-L program must contain a function named main, which takes no arguments and returns no result (the main function is unique in this regard).

The detailed syntax for the MINI-L language is described here. The following table lists the precedence and associativity of all the operators in the MINI-L languages. Operators are listed top to bottom, in descending precedence.

Precedence	Operator	Description	Associativity
0	`()`	Function calls	Left-to-right
1	`[]`	Array subscripting	Left-to-right
2	-	Unary minus	Right-to-left
3	*	Multiplication	Left-to-right
	/	Division	
	%	Remainder	
4	+	Addition	
	-	Subtraction	
5	<	For relational operators <	
	<=	For relational operators <=	
	>	For relational operators >	
	>=	For relational operators >=	
	==	For relational operator ==	
	<>	For relational operator !=	

6	`not`	Logical not	Right-to-left
7	`and`	Logical and	Left-to-right
8	`or`	Logical or	
9	`:=`	Assignment	Right-to-left

MINI-L Example Programs

- mytest.min
- primes.min
