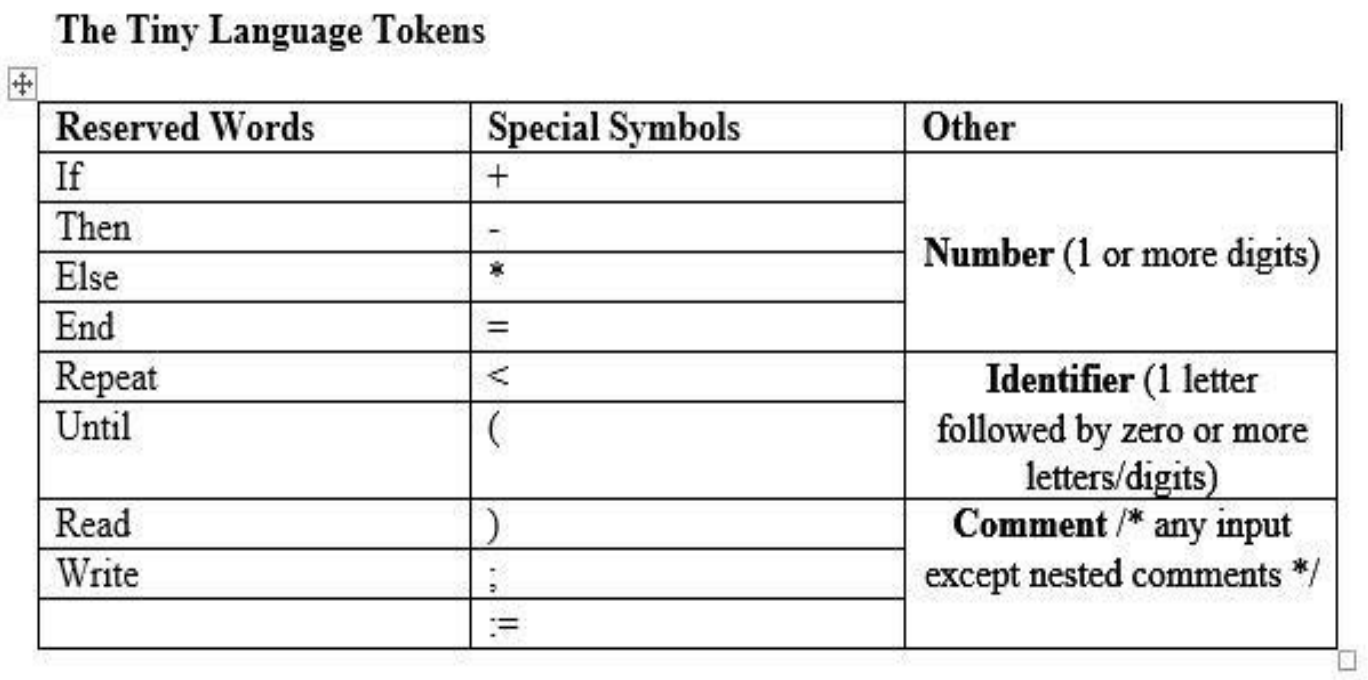
The Tiny Language (Lexical Analyzer)

A program in TINY has a simple structure with the following characteristics (minimum requirements):

- A sequence of statements separated by semicolons.
- No procedures and no declarations
- All variables are integers which are declared by simply assigning values to them (like BASIC)
- Only two control statements which may include statement sequences: o If statement: has an optional else part and must be
  terminated by the word end.
  - Repeat statement
- Read and Write statements that perform input/output
- Multiline comments ({ }) are used for block comments but comments cannot be nested for simplicity. Supporting nested comments is optional.
- Expressions are limited to Boolean and arithmetic expressions.
- Arithmetic expressions may involve: integer constants, variables, parentheses and any of the three integer

operators -, + and * with the usual mathematical properties (presedence and associativity)

- Comparison operators are only: < and =
- Boolean expressions only appear as tests in control statements (no Boolean variables, assignment or I/O).



The Tiny Language Tokens

Reserved Words	**Special Symbols**	**Other**
If	+	**Number** (1 or more digits)
Then	-	
Else	*	
End	=	
Repeat	<	**Identifier** (1 letter followed by zero or more letters/digits)
Until	(	
Read	)	**Comment** /* any input except nested comments */
Write	;	
	:=	

The Tiny Language CFG (Syntax analyzer)

stmt-sequence → stmt-sequence **;** statement | statement

statement → if-stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt

if-stmt →**if** exp **then** stmt-sequence **end | if** exp **then** stmtsequence **else** stmt- sequence **end** repeat-stmt → **repeat** stmt-sequence **until** exp assign-stmt → **identifier** **:=** exp read-stmt → **read identifier** write-stmt → **write** exp

exp → simple-exp comparison-op simple-exp | simple-exp

comparison-op → **<** | =

simple-exp → simple-exp addop term | term
addop → **+** | **-**

term → term mulop factor | factor

mulop → *****

factor → **(**exp**)** | **number** | **identifier**