Syntax Definition Document

Program := Block
Block := (Statement ;)*
Statement := VarDec | Assignment | Input | Print | While | Until | DoRepeat | If |
Unless | FuncDef | FuncCall
VarDec := DataType Ident (= Expression)?
Assignment := Ident = Expression
Input := input Ident
Print := (print | puts | alert) Expression (, Expression)*
While := while Expression do Block end
Until := until Expression do Block end
DoWhile := do Block repeat (while|until) Expression end
If := if Expression then Block (elsif Expression then Block)* (else Block)? End
Unless := unless Expression do Block end
FuncDef := to Ident do Block end
FuncCall := call Ident

Expression := Logical (LogicalOp Logical)*
LogicalExpression := Arithmetic (ComparisonOp Arithmetic)*
ArithmeticExpression := Term (AddOp Term)*
Term := Factor (MulOp Factor)*
Factor := (UnaryOp)? Atom
Atom := IntLit | RealLit | BoolLit | CharLit | StrLit | (Expression) ←literal parentheses

IntLit: (Digit)+
RealLit: IntLit . IntLit
BoolLit: true | false
CharLit: ' [^'] '
StrLit: " ([^\n\r"])* "
Ident: Letter (Letter |Digit | _)*

DataType := int | real | char | string | bool
LogicalOp := and | or
CompOp := is | isnot | < | > | <= | >=
AddOp := + | -
MulOp := * | /
UnaryOp := + | - | not
Letter: A|B|C|D|...|Z|a|b|c|d|...|z
Digit: 0|1|2|3|4|5|6|7|8|9

Comment: # [^\r\n]* \r\n | /* .... */