

<integers> ::= <digit> <integers> | <digit>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<real> ::= <integers> . <integers>

<var\_assgn> ::= var <var\_name> = <i\_expr>

<bool\_var\_assgn> ::= var <var\_name> = <boolean>

Variable names can be anything except for keywords:

<var\_name> ::= !<keywords>

<keywords> ::= var | add | mult | div | mod | true | false | or | not | less | than | equal | greater | to | not | then | end | if | from | to | output | outputs | text | cla | array

<i\_expr> ::= <i\_expr> add <mult\_expr> | <i\_expr> sub <mult\_expr> | <mult\_expr>

<mult\_expr> ::= <mult\_expr> mult <neg\_expr> | <mult\_expr> div <neg\_expr> |

<mult\_expr> mod <neg\_expr> | <neg\_expr>

<neg\_expr> ::= <root> | -<root>

<root> ::= (<i\_expr>) | <var\_name> = <var\_name> | <real> | <integers> | <var\_assgn>

<boolean> ::= true | false

<b\_expr> ::= or(<b\_expr>,<and\_expr>) | and( <and\_expr>,<b\_root>) | <b\_root> |

not(<b\_root>) | <comp> | <var\_name>

<b\_root> ::= <bool> | (<b\_expr>) | <var\_name>

<comp> ::= <i\_expr> less than <i\_expr> |<i\_expr> greater than <i\_expr> |<i\_expr> less than or equal to <i\_expr> |<i\_expr> greater than or equal to <i\_expr> |<i\_expr> equal to <i\_expr> |<i\_expr> not equal to <i\_expr>

<conditionals> ::= if <comp> then <statements> end if | or if <comp> then <statements> end if | or <statement> end if

<break> ::= stop loop now

<loops> ::= <from\_loop> | <for\_each>

<from\_loop> ::= from <var\_assgn> to <comp> increment by <integers> <statements>

end | from <var\_assgn> to <comp> decrement by <integers> <statements> end

<for\_each> ::= for each <var\_name> in <statements> do <statements> end for each

<statements> ::= <statement> | <statement><statements> | <empty>

<statement> ::= <i\_expr> | <conditionals> | <while\_loop> | <from\_loop> | <for\_each>

<print\_to\_output> ::= output(<root>) | outputs(<root>)

<CLA> ::= cmd(<int>) | cmd(<int>) <CLA>

<str\_literals> ::= "<chars>"

<chars> ::= <ASCII Chars>

<arrays> ::= var <var\_name> = array(<integers> <empty> | <integers>, <integers>)

<comments> ::= start comment | end comment | \*<chars>