Lexic.txt:

Alphabet:

a-z

A-Z

0-9

Lexic:

operators: +, -, \*, /, ==, <, <=, >, >=, =

separators: {}, (), ;, space, newline, ", ','

reserved words: int, if, print, readInt, else, while, setNth, getNth

identifier = letter{letter|digit}

letter = "A"|"B"|..|"Z"|"a"|"b"..|"z"

digit = "0"|"1"|..|"9"

integer = integer = "0" | [" + " | " - "] nonzerodigit { "0" | nonzerodigit }

nonzerodigit = "1"|"2"|..|"9"

Tokens.in:

+

-

\*

/

==

<=

>=

<

>

=

{

}

(

)

;

space

newline

"

,

int

if

print

readInt

else

while

setNth

getNth

Syntax.in:

program = {statement}

statement = (var\_statement|arr\_statement|assign\_statement|if\_statement|while\_statement|function\_call\_statement) ";"

var\_statement = "int" identifier\_list

arr\_statement = "int" "[" positive\_number "]" identifier

identifier\_list = identifier ["=" expression] {"," identifier ["=" expression]}

expression = int\_expression

int\_expression = int\_constant | identifier |

int\_expression ("+"|"-"|"\*"|"/") int\_expression |

"(" int\_expression ("+"|"-"|"\*"|"/") int\_expression ")"

expression\_list = expression{"," expression}

assign\_statement = identifier "=" expression

if\_statement = "if" "(" condition ")" "{" {statement} "}" ["else" "{" {statement} "}"]

condition = expression ("=="|"<"|"<="|">"|">=") expression

while\_statement = "while" "(" condition ")" "{" {statement} "}"

function\_call\_statement = function\_name "(" expression\_list? ")"

function\_name = "print" | "set\_nth" | "get\_nth" | "readInt"