**Lexic**

a. Special symbols, representing:

- Operators + - \* / % < <= = \= => > and or not

- Separators [ ] { } , . space

- Reserved words: declare read int string boolean list for while function if otherwise then print

- Comments ~

b. Identifiers: a sequence of letters with no digits

- identifier ::= letter | letter { letter }

- letter ::= “A” | “B” | … | “Z” | “a” | “b” | … | “z” | “+” | “-“ | “\*” | “/” | “%” | “<” | “>” | “=” | “%” | “\*” | “(“ | “)” | “\_” | “{“ | “}” | “[” | “]” | “:” | “ “ ” | “\” | “,” | “.”

c. Constants:

- Integer:

intconst ::= [ “+” | “-“ ] nzDigit { “0” | nzDigit }

digit ::= “0” | “1” | … | “9”

- String:

stringconst ::= “ \” “ { letter | digit | specialSymbol } ” \” “

- Boolean:

booleanconst ::= “true” | “false”

**Tokens**

int string boolean declare read int string boolean list for while function if otherwise then print ~ + - \* / % < <= = \= => > [ ] { } , .

**Sintax**

Program ::= Stmt { Stmt }

Stmt ::= (DeclareStmt | AssignStmt | PrintStmt | IfStmt | WhileStmt | ForStmt) “.”

DeclareStmt ::= type ( identifier [ “[“ intconst “]” ] | AssignStmt )

Type ::= “int” | “string” | “boolean”

AssignStmt ::= identifier “:=” term

Term ::= factor [ operator ( factor | term ) ]

Operator ::= “+” | “-“ | “\*“ | “/“ | “%“

Factor ::= identifier | intconst | stringconst | booleanconst

IfStmt ::= “if” Condition “then” { Stmt } [ “otherwise” { Stmt } ]

Condition ::= term relOperator term

RelOperator ::= “<” | “> “ | “=” | “<=” | “=>” | “\=”

WhileStmt ::= “while” Condition “execute” { Stmt }

ForStmt ::= “for” identifier “:<” ( intconst | expression ) “,” (intconst | expression) “,” intconst “>” {Stmt}

PrintStmt ::= “print” term

ReadStmt ::= “read” type identifier