Mini-language specification

Sintactical rules:

program ::= “GO“ {declList | stmtList} “STOP”

declList ::= declaration | declaration declList

declaration ::= variableDeclaration | constDeclaration

variableDeclaration ::= “var” IDENTIFIER "@" type [“=” expression] “;”

constDeclaration ::= “let” IDENTIFIER "@" type “=” expression “;”

type1 ::= "Bool" | "Int" | “Char” | "String"

arrayDecl ::= “[“ type1 “]”

type ::= type1|arrayDecl

stmtList ::= stmt | stmt stmtList

stmt ::= simplStmt | structStmt

simplStmt ::= assignStmt | ioStmt

assignStmt ::= IDENTIFIER "=" expression “;”

expression ::= expression "+" term | expression “-” term | term | BOOLEAN

term ::= term “*” factor | term “/” factor | term “%” factor | factor

factor ::= “(“ expression “)” | IDENTIFIER | INTEGER

ioStmt ::= “read” “(“ IDENTIFIER “)” “;”| "print" "(" stringExp ")" “;”

stringExp ::= STRING | IDENTIFIER

structStmt ::= ifStmt | whileStmt

ifStmt ::= “if” condition “{” stmtList “}” {elifStmt} [elseStmt]

elseStmt ::= “else” “{” stmtList “}”

elifStmt ::= “elif” condition “{" stmtList “}”

whileStmt ::= "loop" condition "{" stmtList “}”

condition ::= expression RELATION expression

RELATION ::= “<” | “<=” | “=” | “>=” | “>” | ”!=”