

```

1  program = "#" block "#"
2  block = statement ";"
3
4  statement = compoundStatement | declarationStatement | assignmentStatement | ioStatement |
ifStatement | loopStatement
5  compoundStatement = statement ";" statement
6
7  declarationStatement = "let" declaration
8  declaration = type declaree {"," declaree} | declaration "," declaration
9  declaree = identifier | declaree "[" integer "]" | declaree "=" expression
10 type = ("int" | "bool" | "string")
11
12 assignmentStatement = address "=" expression
13 address = identifier | address "[" expression "]"
14
15 expression = listExpression | expression ("+" | "-" | "or") term | term
16 listExpression = "{" "}" | "{" expression {"," expression} "}"
17 term = term ("*" | "/" | "%" | "and") factor | factor
18 factor = "(" expression ")" | relational | address | constant
19 relational = expression comparator expression
20 comparator = "<" | "<=" | "==" | ">=" | ">"
21 constant = integer | boolean | string
22
23 ioStatement = readStatement | writeStatement
24 readStatement = "read" "(" address ")"
25 writeStatement = "write" "(" expression ")"
26
27 ifStatement = "if" "(" expression ")" "then" "{" block "}" ["else" "{" block "}"]
28
29 loopStatement = whileStatement | forStatement
30 forStatement = "for" "(" (declarationStatement | assignmentStatement) ";" expression ";"
assignmentStatement ")" "do" "{" block "}"
31 whileStatement = "while" "(" expression ")" "do" "{" block "}"
32
33 integer = "0" | ["+" | "-"] nonzero {digit}
34 boolean = "true" | "false"
35 string = "" {character} ""
36
37 identifier = (letter | "_") {character}
38 character = letter | digit | "_"
39 letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"
40 digit = "0" | nonzero
41 nonzero = "1" | "2" | ... | "9"

```