```
Alphabet:
    a. Upper (A-Z) and lower case letters (a-z) of the English alphabet

    b. Underline character '_'

    c. Decimal digits (0-9)

Lexic:

    a.Special symbols, representing:

        - operators
            + - * / = < <= == >= and or

        - separators
            [] {} ; space

        - reserved words:
            let bool int string read write if then for while do true false

    b.identifiers

        -a sequence of letters and  digits, such that the first character is a letter;
        the rule is:

            identifier = letter {letter | digit | "_"}
            letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"
            digit = "0" | nonzero
            nonzero = "1" | "2" | ... | "9"


    c.constants

        1.integer:

            integer = "0" | ["+" | "-"] nonzero {digit}

        2.boolean:

            boolean = "true" | "false"

        3.string:

            string = """ {letter | digit | "_"} """
```

```
+
-
*
/
=
<
<=
==
>=
and
or
[
]
{
}
;
let
bool
int
string
read
write
if
then
for
while
do
true
false
```

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