Github: https://github.com/ScortarIrina/SEM5_FLCD/tree/main

Lexic.txt

```
Alphabet:
   A-Z, a-z (uppercase and lowercase letters of the English alphabet)
   0-9 (digits)
   _ (underline character)

a) Special symbols, representing:
   - operators: +, -, *, /, == , <=, <, >, >=, !=, =
   - separators: {}, (), [], ;, "", comma
   - reserved words: read, print, if, else, for, while, int, char, array

b) Identifiers: sequence of letters, digits or “_” such that the first character is a letter
         identifier = letter {letter | digit | “_”}
         letter = “a” | “b” | ... | “z” | “A” | “B” | “D” | ... | “Z”
         digit = “0” | nonZeroDigit
         nonZeroDigit = “1” | “2” | ... | “9”

c) Constants:
         stringConst = " " " {letter | digit | "_" | " "} " " "
         intConst = “0” | [“+” | “-“ ] nonZeroDigit {digit}
```

Syntax.in

```
program = compoundStmt.
compoundStmt = declarationStmt | assignmentStmt | ifStmt | whileStmt | ioStmt.

declarationStmt = simpleType identifier ";" | arrayDecl.
simpleType = “int” | “string” | “char”.
arrayDecl = simpleType identifier “[“ “]” ";".

assignmentStmt = identifier “=” expression ";".
expression = expression + term | expression - term | term.
term = term * factor | term / factor | factor.
factor = “(“ expression “)” | identifier | intConst.

stmtList = stmt | stmt “;” stmtList.
compoundStmt = “{“ stmtList “}”

ioStmt = “read” “(“ identifier “)” ";" | “print” “(“ identifier “)” ";".

ifStmt = “if” “(“ condition “)” compoundStmt|
       “if” “(“ condition “)” compoundStmt “else” compoundStmt.
condition = expression relation expression.
relation = “<” | “<=” | “==” | “!=” | “>=” | “>”.

whileStmt = “while “(“ condition “)” compoundStmt.
```

Token.in

```
+
-
*
/
==
<=
<
>
>=
!=
=
(
)
[
]
{
}
:
;
,
read
write
if
else
while
int
char
array
```

Review (Serbanoiu Stefania)

1. Lexic.txt
   - the alphabet is well-defined
   - various special symbols, treating even *new_line* and *end_of_line*
   - underscore is not considered a possible part of identifiers

2. Syntax.txt
   - the rules are clear and easy to understand
   - array size is taken into consideration
   - conditions between strings and integers are separated

3. Token.in
   - All tokens in the language are specified