Language specification

1.Language definition:
1.1.Alphabet
a. Upper (A-Z) and lower case letters (a-z) of the English alphabet
b. Underline character '_';
c. Decimal digits (0-9);
1.2.Lexic
a.Special symbols, representing:
- operators:
arithmetic: + - * /
assignment: $
relational: < > <= >=
equality: ==
inequality: !=
boolean: !, &&, | |
- separators [ ] { } : ; space
- reserved words:
array char const do else if int for of program read then var while write

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" | "1" | ... | "9"
sign ::= ["+" | "-"]

c.Constants
1.integer - rule:
integer ::= zero | sign digit {(zero | digit)}
2.character
char ::= letter | digit | special
special ::= "-" | "+" | ";" | ...
3.string
constchar ::=" ' " char " ' "
string ::=char{string}
conststring ::= " " " {char} " " "

1.3. Syntax:
The words - predefined tokens are specified between " and ":
Sintactical rules: (file Syntax.in)
program ::= declist ";" cmpdstmt "."
decllist ::= declaration | declaration ";" decllist
declaration ::= identifier ":" type
type1 ::= "bool" | "char" | "int" | "float"
arraydecl ::= "array" "[" nr "]" "of" type1
type ::= type1|arraydecl
cmpdstmt ::= (stmtlist)
stmtlist ::= stmt | stmt ";" stmtlist
stmt ::= simplstmt | structstmt
simplstmt ::= assignstmt | iostmt
assignstmt ::= identifier "$" expression
expression ::= expression "+" term | expression "-" term | term
term ::= term "*" factor | term "//" factor | factor
factor ::= "(" expression ")" | identifier | const
instmt ::= "read" "(" identifier ")"
outstmt ::= "print" "(" identifier ")"

structstmt ::= cmpdstmt | ifstmt | whilestmt
ifstmt ::= "if" "(" condition ")" ":" stmt ["else" stmt]
whilestmt ::= "while" "(" condition ")" ":" stmt
condition ::= expression RELATION expression
RELATION ::= "<" | "<=" | "=" | "!=" | ">=" | ">"