Specification

Alphabet:
a)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
b)
_
c)
0 1 2 3 4 5 6 7 8 9

1. Lexic:
a) Special symmbols, representing
- operators + - * / == < <= = >=
- separators ; space [] : {}
- reserved words:
  array char const in do else if int of program readk readf then while write_file console_print

b) identifiers
identifier ::= letter | letter {(letter | digit)}

letter == "A" | "B" | . ..| "Z" | "a" | "b" | . ..| "z"

digit == "0" | "1" |...| "9"

c) constants
1.Int
number ::= nonZeroDigit{digit}
nonZeroDigit ::= "1" |...| "9"
sign ::= + | -
zero ::= 0
int ::= [sign] number | zero
2. Char
character ::= ‘symbol’
3.String
string ::= “{symbol}”
symbol ::= A | B | … | Z | a | b | … | z |0 | … | 9 | space | # | & | ^ | %
4.Bool
bool ::= true | false

2. Syntax:
program ::= "begin" "main" "{" decllist ";" cmpdstmt "end" "main" "}"
decllist ::= declaration | declaration ";" decllist
declaration ::= IDENTIFIER "->" type
type1 ::= "int"|"string"
arraydecl ::= "array" "[" const "]" "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 | constant | []
iostmt ::= "READK" | "READF" | "READ" "(" IDENTIFIER ")" | "print_console" "[" IDENTIFIER|constant "]" | "
WRITE" "[" IDENTIFIER|constant "]"
structstmt ::= cmpdstmt | ifstmt | whilestmt | forstmt
ifstmt ::= "IF" "[" condition "]" "THEN" stmt "ELSE" "[" stmt "]"
whilestmt ::= "WHILE" "[" condition "]" stmt
forstmt ::= "FOR" "[" assignstmt ";" condition ";" expression "]"
condition ::= expression RELATION expression
RELATION ::= "<" | "<=" | "=" | "<>" | ">=" | ">"
```