LAB2

# Alphabet

* 1. Upper and lower case letters of the English alphabet: A-Z, a-z
  2. Decimal digits: 0-9
  3. Underline character: \_

# Lexic

## Special symbols, representing:

* + 1. operators: + (addition)

- (subtraction)

\* (multiplication)

/ (division)

% (modulo)

>, >=, <, <=, ==, != (for comparison)

= (assignment)

operator := “+” | “-” | “\*” | “/” | “%” | “>” | “>=” | “<” | “<=” | “==” | “!=” | “=” | “&&”

* + 1. separators: >> (used for reading)

<< (used for writing)

:

;

space

{ }, [ ], ( )

separator := “>>” | “<<” | “:” | “;” | “"” | “'” | “space”|

“{” | “}” | “[” | “]” | “(” | “)”

* + 1. reserved words: read, write, if, else, while, for, in, range, Integer, String,

Char, main

reserved\_words := “read” | “write” | “if” | “else” | “while” | “for” | “in” | “range” | “Integer” | “String” | “Char” | “main”

* 1. **Identifiers:** a sequence of letters and digits, starting with underline character (\_) and followed by a letter; the rule is:

identifier := “\_”letter | “\_”letter{letter | digit} letter := “A”|”B”|..|”Z”|”a”|”b”|..|”z”

digit := “0”|”1”|..|”9”

## Constants:

* + 1. integer rule:

integer := “0” | [“+” | “-”]nz\_digit{digit}

nz\_digit := “1”|”2”|..|”9”

digit := “0”|”1”|”2”|..|”9”

letter:= “A”|”B”|..|”Z”|”a”|”b”|..|”z”

char:=’letter’ | ’digit’

string:=”{letter | digit}”

constant:=string|char|integer

* + 1. char rule:

char := ‘letter’ | ‘digit’

letter := “A”|”B”|..|”Z”|”a”|”b”|..|”z” digit := “0”|”1”|..|”9”

* + 1. string rule:

string := “{letter|digit}”

letter := “A”|”B”|..|”Z”|”a”|”b”|..|”z” digit := “0”|”1”|..|”9”

# Syntax

program := “main” “(” “)” “{” statement “}”

statement := {declaration\_statement | assignment\_statement | if\_statement | for\_statement | while\_statement | read\_statement | write\_statement “;”}

declaration\_statement := variable\_declaration\_statement | array\_declaration\_statement variable\_declaration\_statement := identifier\_list “:” type “;”

array\_declaration\_statement := identifier\_list “:” type “[” “]” “;”

identifier\_list := identifier {“,” identifier} type := “integer” | “string” | “char”

expression := int\_expression | string\_expression | char\_expression

int\_expression := integer | int\_expression symbol int\_expression

symbol := “+” | “-” | “\*” | “/” | “%”

string\_exression := string | string + string\_expression

char\_expression := char

assignment\_statement := identifier “=” (identifier | expression) “;”

if\_statement := “if” “(” condition “)” “{” statement “}” [“else” “{” statement “}”]

condition := expression relation expression {&& expression relation expression}

relation := “>” | “>=” | “<” | “<=” | “==” | “!=” | “&&”

while\_statement := “while” “(” condition “)” “{” statement “}”

for\_statement := “for” identifier “in” identifier |

“for” “(“ identifier “=” constant “;” identifier “<” constant “;” identifier “=” identifier “+” constant “)” “{” statement “}”

read\_statement := “read” “>>” identifier {“>>” identifier} “;”

write\_statement := “write” “<<” (identifier|constant) {“<<” identifier} “;”