Lab 1b . Minilanguage specification

Alphabet:

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

b. Underline character '_'

c. Decimal digits (0-9)

d. Special characters ($, #, &, |, /, !, %, *)

Lexic:

a.Special symbols, representing:

- operators : Arithmetic: +, - , %, /, *

Assignment: =

Relational: ==, >, >=, <, <=, !=

- separators [ ] , { }, (), ; , : , space

- reserved words: check, then, for, while, in, int, char, return, read, print, void , program

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"

c.constants

1.integer

integer = “0” | [“+ ” | “-”]non_zero_digit{digit}

non_zero_digit = “1” | “2” | … | “9”

digit = “0” | non_zero_digit

2.character

character ='letter'|'digit'

3.string

string = “{char_seq}”

char_seq = letter | digit

Syntax:

Syntactical rules:

program = “program” “{” stmtSeq “}”

stmtSeq = stmt | stmt “;” stmtSeq

stmt = declaration | simpleStmt | compStmt | exitStmt

declaration = “def” type IDENTIFIER “;”

type = normalType | arrayType

normalType = “int” | “char” | “bool”

arrayType = normalType “[“ ”]”

simpleStmt = assignStmt | readWriteStmt

assignStmt = IDENTIFIER “=” exprs “;”

exprs = term | exprs op exprs | “(” exprs op exprs “)”

term = IDENTIFIER | CONSTANT

op = “*” | “+”| “-” | “/” | “%”

readWriteStmt = readStmt | writeStmt

readStmt = “read” “(” IDENTIFIER “)” “;”

writeStmt = “print” “(” IDENTIFIER “)” “;” | “print” “(” CONSTANT“)” “;”

compStmt = checkStmt | forStmt | whileStmt

checkStmt = “check” “(” condition “)” “then” “{” stmtSeq “}” | “check” “(” condition “)” “then” “{” stmtSeq “}” “else” “{” stmtSeq “}”

forStmt = “for” IDENTIFIER “in” “[“ exprs “,” exprs “]” “{“ stmtSeq “}”

whileStmt = “while” “(” condition “)” “{” stmtSeq “}”

condition = exprs RELATION exprs

RELATION = “<” | “>” | “<=” | “>=” | “==” | “!=”

exitStmt = “break” “;”

Tokens

identifier
constant
return
check
else
while
for
int
char
print
read
program
bool
=
==
>=
<=
<
>
!=
+
-
/
%
*
,
;
(
)
{
}

[
]
!