Alphabet:

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

c. Decimal digits(0-9);

Lexic:

a.Special symbols:

-operator: plus, minus, smaller, bigger, equal, times, modulo, divide

-separator:[], ; , space

-reserved words: loop, do, go, close,print

b.Identifiers

-a sequence of letters and digits, such that the first character is a letter; the rule is:

identifier ::= letter | letter{letter}{digit}

letter ::= "A" | "B" | . ..| "Z"

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

c.Constants

1.Integer:

nconst:="+"n|"-"n|n

n:=digit{no}

2.Character:

character:='letter'|'digit'

3.String:

constchar:="string"

string:=char{string}

char:=letter|digit

Syntax:

program ::= "def" decllist ";"

decllist ::= declaration | declaration decllist

declaration ::= type IDENTIFIER

type1 ::= "bool" | "char" | "int" | "float"

arraydecl ::= type IDENTIFIER "[" nr "]"

cmpdstmt ::= "go" stmtlist "close"

stmtlist ::= stmt | stmt stmtlist

stmt ::= simplstmt | structstmt

simplstmt ::= assignstmt | iostmt

assignstmt ::= IDENTIFIER "->" expression

expression ::= expression "plus" | minus | times | modulo | divide

iostmt ::= "read" | "print" IDENTIFIER

structstmt ::= cmpdstmt | ifstmt | whilestmt | forstmt

ifstmt ::= "if" condition "go" stmt ["ELSE" stmt]

whilestmt ::= "while" condition "go" stmt

forstmt ::= "loop" assignstmt ";" codintion ";" ";" statement

condition ::= expression RELATION expression

RELATION ::= "smaller" | "smallerequal" | "equal" | "biggerequal" | "bigger"

Token

identifier ----- 0

const ----- 1

(----- 3

) ----- 4

{ ----- 5

} ----- 6

[----- 7

] ----- 8

~ ----- 9

. ----- 10

+ ----- 11

- ----- 12

* ----- 13

/ ----- 14

or ----- 15

and ----- 15

show ----- 16

read ----- 17

stop ----- 18

if ----- 19

elseif ----- 20

while ----- 21

for ----- 22

var ----- 23

integer ----- 24

boolean ----- 25

vector ----- 26

PERROR

```
int a[1, 3, 5, 7, 9]
loop i is 0
   a[i] is -1
close
```