

Alphabet:

a-z

A-Z

0-9

—

1.Lexic:

a.Special symbols

-operators: +, -, \*, /, <-, <, <=, >, >=

-separators: [], {}, //, :, \*

-reserved words:

array, con, if, else, do, while, var, read, write, for

b.identifiers

-a sequence of digits:

identifier ::= digit|digit{digit}

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

c.constants

1.int:

nrconst:="+nr"|"nr|nr

nr:=digit{nr}

2.char:

char:='letter'|'digit'

3.str:

constchar:="string"

string:=char{string}

program ::= "start" cmpStmt

statement ::= /decl | assigStmt | ifStmt |

whileStmt | returnStmt | forStmt| ioStmt/

decl::="/"IDENTIFIER"/" : type

type::="BOOL"|"CHAR"|"INT"|"REAL"|"STR"

arrayDecl ::= "array" " " type " " IDENTIFIER "[" "]"

stmtList ::= stmt | stmt "\*" statement\_list

cmpStmt ::= "{" statement\_list "}"

expression ::= expression + term | expression - term | term

term ::= term \* factor | term / factor | factor

factor ::= "/" expression "/" | IDENTIFIER | CONST

ioStmt ::= "read" "/" IDENTIFIER "/" | "write" "/" IDENTIFIER "/" | "write" "/" CONST "/"

assignStmt ::= IDENTIFIER "<-" expression

ifStmt::="if" "/" condition "/" cmpStmt| "if" "/" condition "/"cmpStmt "else" cmpStmt

whileStmt ::= "while" "/" condition "/" cmpStmt

```
doWhileStmt ::= "do" cmpStmt "while" "/"condition"/"  
returnStmt  ::= "return expression"  
forStmt     ::= "for" forCond cmpStmt  
forCond     ::= "/" "int" assignStmt ";" condition ";" assignStmt ")"
```

```
condition ::= expression relation expression  
relation  ::= "<" | "<=" | "==" | "!=" | ">=" | ">"
```

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  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
+  
-  
\*  
/  
==  
<=  
<  
>  
>=  
!=  
=  
(  
)  
[  
]  
{  
}  
:  
;  
,  
space  
read  
write  
if  
else  
for  
while  
do  
int  
str  
real  
char  
return  
start  
array