Lexic.txt

Alphabet:

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

b. Decimal digits (0-9);

c. Non alphanumeric characters: '-', '_';

Lexic:

a.Special symbols, representing:

- operators + - * / = == += -= *= /=

- separators ( ) [ ] { } ; space

- reserved words:

int char if else for while read 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" | ... | "z" | "A" | "B" | ... | "Z"

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


   c.constants

1.integer - rule:

   int:="+"no|"-"no|no

        nonZeroDigit:= "1" |...| "9"

   no:=nonZeroDigit | no{digit}

2.character

  character:=""letter""|""digit""

3.array

   array:=digitArray | charArray

   charArray:=char | charArray{char}

   digitArray:=digit | digitArray{digit}


Syntax.txt
```

```
program ::=  stmtlist

declaration ::= type IDENTIFIER";"

type1 ::= "char" | "int"

arraydecl ::= type1 "[" nr "]" IDENTIFIER

type  ::= type1|arraydecl

stmtlist ::= stmt";" | stmt ";" stmtlist

stmt ::= simplstmt | structstmt | declaration

simplstmt ::= assignstmt | iostmt

assignstmt ::= IDENTIFIER "=" expression |
IDENTIFIER "+=" expression |
IDENTIFIER "-=" expression |
IDENTIFIER "*=" expression |
IDENTIFIER "/=" expression |

expression ::= expression "+" term |
expression "-" term |
expression "*" term |
expression "/" term |
term
```

term ::= IDENTIFIER | charArray | no

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

"print(" charArray")" | "print(" no ")"

structstmt ::= stmtlist | structstmt ifstmt | structstmt whilestmt

ifstmt ::= "if(" condition ") {" structstmt "} else {" structstmt "}"|

"if(" condition ") {" structstmt "}"

whilestmt ::= "while(" condition ") {" stmt "}

condition ::= expression RELATION expression

RELATION ::= "<" | "<=" | "=" | "!=" | ">=" | ">"

Tokens.txt

int

char

=

==

-=

+=

/=

+

-

*

/

if

else

for

while

print

read

( )

[ ]

{ }

;

Space

P1

Check if a given number is a perfect square.

```
int n;
read(n);
int i = 0;
char[3] da;
da = "DA";
char[3] nu;
nu = "NU";
while(i*i < n) {
	i += 1;
}

if(i*i == n) {
	print(da);
}
else {
        print(nu);
}
```

P2

Compute the gcd of 2 given numbers.

```
int a;
int b;
int r;
int ca;
int cb;
read(a);
read(b);
r = a % b;
ca = a;
cb = b;
while(b != 0) {
        a = b;
        b = r;
        r = a%b;
}
print(a);
```

P3

Compute the sum of n given numbers, where n is given.

```
int n;
read(n);
int x;
int s
s = 0;
int i;
i = 0
while(i < n) {
	read(x);
	s += x;
	i += 1;
}
print(s);
```

P3err

Compute the sum of n given numbers, where n is given.

```
int n -- syntax error, unknown instantiation
read(n);
int x;
s = 0; -- syntax error, can not fin s
int i;
i = 0
while(i < n) {
	read(x);
	s += x;
	i += 1;
}
print s; -- syntax error, print is a function
```