Ravasz Tamás

<https://github.com/RavaszTamas/FormalLanguagesLaboratories>

Lexic:

**Operators**:

kapja (assignment)

osszead (add)

szoroz(multiply)

nemegyenlo ( != )

egyenlo (==)

modulo

**Keywords**:

egesz (integer)

karakterlanc (string)

**Reserved words:**ha ( if )

**Separators**: space , : , ; , Kezd (start), Vege(end)

<constant>::= 0 | 1

<letter>::= a | b | c | d

<identifier> ::= <letter> | <letter><seq\_of\_letter>

<seq\_of\_letter> ::= <letter><seq\_of\_letter> | <letter>

<input\_statement>::= olvas <identifier>

<output\_statement> ::= kiir <identifier>

P1: max from three numbers

Kezd

egesz a,b,c,d;

olvas\_a;

olvas\_b;

olvas\_c;

ha a nagyobb b:

d kapja a:

kulonben:

d kapja b

ha c nagyobb d:

d kapja c

kiir d

Vege

P2 concatenate:

Kezd

karakterlanc result\_string,first\_string,second\_string;

beolvas first\_string;

beolvas second\_string;

result\_string kapja first\_string osszead second\_string;

kiir(first\_half);

kiir(second\_half);

Vege

P3 check\_if\_relative\_prime:

Kezd

egesz a,b, c;

beolvas a;

beolvas b;

amig a is not 0 :

ures kapja second\_number;

second\_number kapja first\_number modulo second\_number;

first\_number kapja ures;

amig vege

erdemeny first\_number

ha(gcd(a,b) egyenlo 1):

kiir “relative primes”

kulonben

kiir “not relative primes”

Vege

P1err: calculate interest rates

Kezd

egesz $money;

dupla 1yeartime = 365;

dupla #\_interest\_rate\_#;

read $money ;

beolvas #\_interest\_rate\_# ;

kiir $money szoroz ( 1 osszead #\_interest\_rate\_# szoroz 1yeartime);

Vegez