

Ravasz Tamás

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

Lexic:

Operators:

kapja (assignment)

osszead (add)

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