ISETL-Harjoitus 2, 1999

Tehtävä 1. Triviaali syt-algoritmi ilman funktiota `onko_tekija`

Seuraava funktio (vrt. syt demoissa 1) etsii kahden luvun suurimman yhteisen tekijän käyttämättä hidastavaa funktionkutsua. Kirjoita se ja aja ISETLissä.

```
>  syt := func(m,n);
>>  local s; s := 1;
>>  for i in [1..m] do
>>  if ((m mod i = 0) and (n mod i = 0)) then s := i; end if;
>>  end for;
>>  return s;
>>  end func;
```

Kokeile funktiota luvuilla, joista toinen on paljon suurempi kuin toinen, kokeile molemmin päin. Entä negatiiviset luvut? Korjaa funktio oikein toimivaksi.

Tehtävä 2. Tutkiskelua

a) Tutki `syt`-funktion avulla seuraavien lukujen suurinta yhteistä tekijää: 36 ja 96, 36 ja (96+36), 36 ja (96+2·36), 36 ja (96−36), 36 ja (96 − 2·36), ja yleisesti 36 ja $96+k36$, $k \in \mathbf{Z}$. Testaa havaintoasi myös jollakin muulla lukuparilla.
b) Onko jokin luvuista $96 + k36$, $k \in \mathbf{Z}$, sama kuin $96 \bmod 36$? Entä onko jokin luvuista $a+kb$, $k \in \mathbf{Z}$, sama kuin $a \bmod b$? Muotoile sen jälkeen havaintosi matemaattisella kielellä.

Tehtävä 3. Rekursiivinen algoritmi

Mitä seuraava rekursiivinen, eli itseään kutsuva funktio `syt2` tekee (lisää vaikkapa tulostava käsky väliin)? Toimiiko se aina? (`while [ehto] do` on taas silmukka, joka pyörii niin pitkään kuin `ehto` on voimassa)

```
>  syt2 := func(a,b);
>>    while (b /= 0) do
>>    return syt2( b, a mod b);
>>    end while;
>>  return a;
>>  end func;
```

Tehtävä 4. Lineaarikombinaatioesitys

Kirjoita funktio, jonka syötteenä on kaksi kokonaislukua a ja b ja joka etsii lukuja x ja y, joille

$$\mathrm{syt}(a, b) = ax + by.$$

Funktiosi tulee palauttaa lista (eli TUPLE) tällaisista lukupareista $[x, y]$. Huom! Sinun on pakko rajoittaa luvut x ja y jollekin välille, esim. `x,y in [-50..50]`, koska ISETL ei pysty käsittelemään äärettömiä joukkoja. Näin ollen funktiosi ei välttämättä toimi kaikille syötteille.

Toisin sanoen, funktiosi etsii lukujen a ja b suurimmalle yhteiselle tekijälle esityksen tai esityksiä lukujen a ja b *lineaarikombinaatio(i)na*.
Kun funktiosi on valmis, testaa sitä ainakin luvuilla 4 ja 5, 26 ja 4, -5 ja 10. Kuinka monta lineaarikombinaatiota funktiosi löytää?

Tehtävä 6. Kvanttoreista (jäänee kotitehtäväksi)

a) Kirjoita seuraavat rivit ohjelmalle, ja yritä arvata, mitä kukin rivi tarkoittaa ennen kuin suoritat sen.

```
> exists n in {3,5,4,91} | even(n);
> exists n in {3,5,7,13} | even(n);
> forall n in {1..100} | n > 0;
> forall n in {1..100} | even(n);
> Z20 := {0..19};
> forall x in Z20 | (x+0) mod 20 = x;
> forall x in Z20 | (x+3) mod 20 = x;
> exists g in Z20 : (g+5) mod 20 = 0;
> exists g in Z20 : (g+4) mod 20 =0;
> choose g in Z20 : (g+4) mod 20 = 0;
> forall x in Z20 : (exists g in Z20 : (g+x) mod 20 = 0);
> exists e in Z20 : (forall x in Z20 : (x+e) mod 20 = x);
> choose e in Z20 : (forall x in Z20 : (x+e) mod 20 = x);
```

b) Kirjoita ISETL-koodi, jolla tarkistat, ovatko seuraavat väittämät totta vai eivät:
1) Jokainen joukon $\mathbf{Z}_{20}$ alkio on parillinen.
2) Jokin joukon $\mathbf{Z}_{20}$ alkio on pariton.
3) Joukossa $\{1, 3, 5, 7\}$ on alkio, jolla kerrottuna kukin joukon $\{1, 3, 5, 7\}$ alkio pysyy ennallaan.
4) Jokaiselle joukon $\mathbf{Z}_{20}\backslash\{0\}$ alkiolle löytyy alkio joukosta $\mathbf{Z}_{20}\backslash\{0\}$ niin, että kyseisten alkioiden tulo mod 20 on 1.

Kvanttoreista $\forall$ ja $\exists$

Matematiikassa käytetään usein lauseita, joissa esiintyy ajatus 'kaikille' tai 'on olemassa', ts. 'löytyy ainakin yksi sellainen, että ... ' Näille kahdelle ajatukselle on omat merkkinsä: $\forall$ luetaan 'kaikille' tai 'kaikilla', ja $\exists$ tarkoittaa 'on olemassa'. ISETLissä näiden tällaisten lauseiden rakenne muistuttaa joukkojen muodostamista: esimerkiksi

```
> forall x in Z20 : (x+0) mod 20 = x;
```

on matematiikan ilmaus: $\forall x \in \mathbf{Z}_{20} : (x + 0) \bmod 20 = x$.
Kvantifiointilauseessa on kaksi osaa: esimmäinen osa alkaa avainsanalla `forall` tai `exists` ja jatkuu määrittelyjoukon antamisella (`x in Z20`). Toinen osa (:-merkin jälkeen) on looginen lauseke, eli lauseke, jonka arvo on joko tosi tai epätosi. ISETL palauttaa `forall` tai `exists` lauseesta totuusarvon `true` tai `false`.
Lisäksi ISETLissä on olemassa operaatio `choose`, jonka syntaksi on täsmälleen samanlainen kuin operaation `exists` syntaksi. Se todella etsii kyseisen ehdon toteuttavan alkion annetusta joukosta. Näin ollen se ei palauta totuusarvoa, vaan joukon alkion.