Tartalomjegyzék

Megfordítás

Írjunk egy függvényt, aminek bemenete egy lista és a lista elemeinek sorrendjét megfordítja.
Például:

```
megfordít([1,2,3,4,5])
[5,4,3,2,1]
```

Oszthatóság

1.

Írjunk egy 2 paraméter? függvényt osztható() néven a következ? módon: Az els? bemenete legyen egy lista, a második pedig egy természetes szám. A függvény térjen vissza a lista azon elemeinek listájával, amelyek oszthatók ezzel a természetes számmal.
Például:

```
osztható(list(range(30,50)),7)
[35, 42, 49]
```

2.

Definiáljunk egy osztók() függvényt, ami egy természetes szám valódi osztóinak listáját adja vissza.

Listák

1.

Írjunk egy függvényt dupla() néven, ami bemenetnek 1 listát kap és visszatér **True**-val, ha van benne olyan elem, ami legalább kétszer szerepel, egyébként visszatér **False**-szal.

2.

Írjunk egy függvényt listatöbbször() néven, aminek 2 paramétere egy lista és egy természetes szám. listatöbbször(l,n) térjen vissza **False**-szal, ha a listában nincs legalább *n* különböz? elem, egyébként pedig **True**-val.

3.

Definiáljunk egy függvényt rendezett_e() néven, aminek egy paramétere egy egész számokból álló lista, és eldönti, hogy rendezett-e a lista. Ha rendezett, akkor térjen vissza **True**-val, egyébként **False**-szal. Ha listában egy elem is többször szerepel, akkor térjen vissza **None**-nal.

Válogatás

Adott egy számokat tartalmazó L listánk, írjunk programot, mely két listába válogatja L elemeit, negatívakat az egyikbe, nem negatívakat a másikba.

```
L = [-1, 2, 5, -2, 3, -4, -5, 2, -2, 0, 5, 5, 6, 3, -3]
```

Prímfaktorizáció

Írjunk egy függvényt prím_faktorizáció() néven, aminek bemenete egy természetes szám és egy olyan listával tér vissza, amiben párok vannak. A pár els? tagja adja vissza, hogy melyik prím szerepel a faktorizációban, a második tagja pedig, hogy milyen hatvánnyal szerepel.
(Tipp: Felhasználható erre a célra a max_exp() függvény.)
Például:

```
prime_decomp(90)
[(2, 1), (3, 2), (5, 1)]
```

Lista a listában

Definiáljuk egy függvényt, aminek 2 bemenete van, egy lista, amiben listák vannak és a másik paramétere egy természetes szám. A listahossz(l,n) függvény adja vissza az l listában található azon listák számát, melyek n hosszúak.

Például:

```
listahossz([[1,2,3],[2,3],[1,2,6,5],[2,3,2]], 3)
2
```

Szél?értékek

Írjuk meg a minimum() és maximum() függvényt, aminek a bemenete egy lista és kiemenete a lista legkisebb, illetve legnagyobb eleme.

Legnagyobb, legkisebb

Írjunk egy két paraméteres függvényt `szelsoertek` néven, els? paramétere legyen: *l*, ami egy számokat tartalmazó lista, a második pedig egy **True** vagy **False** érték

A függvény térjen vissza a lista legnagyobb elemek indexeivel, ha a második paraméter **True**, egyébként pedig legkisebb elemének indexeivel.

Tuple

Definiáljunk egy függvényt lookup() néven, aminek 2 argumentuma van. A második argumentuma egy lista, ami 2 hosszú tuple-ket tartalmaz, az els? argumentum pedig a kulcs. A lookup(kulcs, lista) hívás térjen vissza az els? olyan tuple második tagjával, aminek az els? tagja megegyezik a kulcs bemenettel. Ha nincs ilyen tuple a listában, akkor térjen vissza None-nal.

Legnagyobb közös osztó

Definiáljuk az lnko() függvényt, aminek paramétere két természetes szám és visszatér a legnagyobb közös osztójukkal. (Ehhez felhasználható a prím_faktorizáció() függvény.)