el?z? fel következ?

Tartalomjegyzék

Feladatok

Faktoriális

Adjuk meg a bemeneti szám faktoriálisát! (Bemenet: *n*, kimenet *n!*)

Tökéletes számok

Írjunk programot, mely bekér egy pozitív egész számot és leellen?rzi, hogy tökéletes szám-e.

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]
```

Lista els?

A python egy típusa amivel gyakran fogunk találkozni, a `list`. A lista sok szempontból hasonlít a karakterláncra, csak elemei nem bet?k, hanem tetsz?leges python dolgok. Egyel?re számok lesznek a listák elemei.

Írjunk python függvényt, ami visszaadja a lista els? elemét, ha létezik (tehát ha van legalább egy eleme a listának). A függvény neve legyen `lista_elso`, egy paramétere legyen: *l*, a lista. A függvény a lista els? elemével térjen vissza ha létezik, és a speciális `None` értékkel ha nem.

Lista vége

Írjunk függvényt ami csak a lista utolsó valahány elemét tartja meg!

A függvény neve legyen `lista_veg`, két paramétere legyen: *l*, a lista *db*, a megtartandó elemek száma. Ha van az *l* listának legalább *db* eleme, akkor az utolsó *db* darabbal térjen vissza a függvény, egy *db* elem? listában. Ha *l*-nek kevesebb mint *db* eleme van, akkor a függvény `None`-nal térjen vissza.

Szín komponens

A számítógépek a színeket általában 3 komponensben tárolják. A leggyakrabban használt módszert RGB kódolásnak hívják, ami a "http://wiki.math.bme.hured"http://wiki.math.bme.hu, "http://wiki.math.bme.hugreen"http://wiki.math.bme.hu és "http://wiki.math.bme.hublue"http://wiki.math.bme.hu szavak kezd?bet?jéb?l áll össze, mert a szín komponensei ilyenkor a piros, zöld, kék sorrendben vannak megadva.

Írjunk függvényt ami név alapján ki tudja választani a megfelel? komponenst.

A python függvény neve legyen `szin_komponens`, két paramétere legyen:

- *szin*, egy három elem? lista, ami a három komponens er?ségét tartalmazza három 0 és 1 közötti valós számként
- *komponens*, a "http://wiki.math.bme.hupiros"http://wiki.math.bme.hu, "http://wiki.math.bme.huzold"http://wiki.math.bme.hu és "http://wiki.math.bme.hukek"http://wiki.math.bme.hu karakterláncok egyike
- A függvény térjen vissza a *komponens*-nek megfelel? szín komponens er?ségével, tehát a lista megfelel? elemével.

Sorszámok

Megvan egy versenyen induló versenyz?k listája. Meg akarjuk ?ket számozni 1-t?l indulva.

Írjunk függvényt, ami visszaadja a szükséges sorszámok listáját, hogy ki tudjuk a sorszámokat nyomtatni.

A függvény neve legyen `sorszamok`, egy paramétere legyen: *indulok*, a versenyz?k listája. Ha az *indulok* listának N eleme van, akkor a függvény egy olyan listával térjen vissza, ami a számokat 1-t?l N-ig tartalmazza.

Naplóból

Egy gimis tanár úgy szokta eldönteni hogy ki feleljen az óráján, hogy sorsol egy számot *0* és az *osztály létszáma-1* között, majd az felel aki a naplóban annyiadik helyen van.

Írjunk függvényt ami megmondja hogy ki fog felelni, a diákok listájából, és a sorszámból amit kisorsolt a tanár. Az egyetlen gond hogy a mi listánk nincs névsorba rendezve mint a napló.

A függvény neve legyen `naplobol`, két paramétere legyen:

- *nevek*, egy lista ami az osztály diákjait tartalmazza, de nincs sorba rendezve
- *sorszam*, egy szám, hogy a sorbarendezett listából hányadikdiák fog felelni. Ha ez a szám 0, akkor a névsorban els? fog felelni, és így tovább.

A függvény a felel? nevével térjen vissza.

Legnagyobb

Írjunk python függvényt, ami megmondja hogy hányadik egy lista legnagyobb eleme.

A függvény neve legyen `legnagyobb`, egy paramétere legyen: *l*, ami egy számokat tartalmazó lista.

A függvény térjen vissza a lista legnagyobb elemének indexével. Például az `[3, 2, 1]` listára a 0 számot adja, mert a nulladik a legnagyobb.

Ezt a m?veletet szokás `argmax`-nak is nevezni.

Udvarias

A bemenetünk egy lista: *l*, ami n?i neveket tartalmaz. Alkossunk minden névb?l egy udvarias megszólítást úgy, hogy elé tesszük hogy "http://wiki.math.bme.huMs."http://wiki.math.bme.hu. Így pl. abból hogy "http://wiki.math.bme.huAlice"http://wiki.math.bme.hu, legyen "http://wiki.math.bme.huMs. Alice"http://wiki.math.bme.hu. Az így módosított listát adjuk vissza visszatérési értékként.

Szorzó

A bemenetünk egy lista: *l*, ami számokat tartalmaz, és egy szám: *k*. Adjuk vissza azt a listát, amit úgy kapunk, hogy *l* minden elemét megszorozzuk *k*-val.

Neptun el?bb

Az egyetemen a különféle számonkérések eredménye sokszor Neptun kód alapján van közölve, nem a név alapján. Ilyenkor, f?leg egy hosszú listában, nehéz lehet megkeresni a saját eredményünket. Írjunk egy függvényt, ami megmondja egy Neptun kódok alapján rendezett listában, a mi Neptun kódunk és az éppen látott Neptun kód alapján, hogy a mi sorunk el?bb van-e a táblázatban vagy sem.

Írjunk python függvényt, neve legyen `neptun_elobb`, két paramétere legyen:

- *mienk*, a mi Neptun kódunk
- *aktualis*, az éppen most látott Neptun kód.

A függvény a speciális `True` azaz igaz értékkel térjen vissza ha a mi kódunk van el?bb, és a `False`, azaz hamis értékkel, ha kés?bb.

Cloudcoder

1. ismetles
2. kiejtes
3. buvos_negyzet
4. szorzotabla
5. pascal
6. cserebere

el?z? fel következ?