

el?z? fel következ?

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1 Válogatás
 - ◆ 1.2 Lista els?
 - ◆ 1.3 Lista vége
 - ◆ 1.4 Szín komponens
 - ◆ 1.5 Sorszámok
 - ◆ 1.6 Naplóból
 - ◆ 1.7 Legnagyobb
 - ◆ 1.8 Udvarias
 - ◆ 1.9 Palindrom
 - ◆ 1.10 Szorzó
 - ◆ 1.11 Neptun el?bb
- 2 Trükkösebb feladatok
 - ◆ 2.1 Összefésüléses rendezés
 - ◆ 2.2 Részösszeg=0
 - ◆ 2.3 Listák listája

Feladatok

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 tets?leges python dolgok. Egyel?re számok lesznek a listák elemei.

Írjunk python függvényt, ami visszaadja a lista első 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 első 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.huned" "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 kezdőbetűjéből áll össze, mert a szín komponensei ilyenkor a piros, zöld, kék sorrendben vannak megadva.

Írjunk függvényt ami név alapján ki tudja választani a megfelelő komponenset.

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

- `szin`, egy három elemű lista, ami a három komponens értékét 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 megfelelő szín komponens értékével, tehát a lista megfelelő elemével.

Sorszámok

Megvan egy versenyen induló versenyzők listája. Meg akarjuk őket számozni 1-től 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 versenyzők listája. Ha az `indulok` listának `N` eleme van, akkor a függvény egy olyan listával térjen vissza, ami a számokat 1-től `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:

Lista első

- *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 műveletet 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 értéként.`

Palindrom

Írjunk egy függvényt, melynek bemenete egy lista, és attól függ?en ad vissza `True` vagy `False` értéket, hogy az adott lista palindrom-e.

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.

Trükkösebb feladatok

Összefésüléses rendezés

Legyen adott két rendezett listánk l_1 és l_2 . Fésüljük össze az elemeiket egy l_3 listába úgy, hogy az új lista is rendezett legyen. Erre megoldást nyújt az összefésüléses rendezés, melynek lényege, hogy párosával összehasonlítjuk sorrendben a két lista elemeit és mindig a kisebbet tesszük el?ször az összefésült listába.

Segítség: [Összefésüléses rendezés](#) 18.1. elegend? a feladat megoldásához.

Részösszeg=0

Legyen adott egy l listánk, határozzuk meg, hogy összesen hány olyan összefügg? részintervalluma van, melyre teljesül, hogy a benne lév? számok összege 0.

Listák listája

Legyen adott egy L listánk, melynek elemei is listák. Írjunk egy olyan megfordit (lista) függvényt, mely visszatér egy olyan listával, melyben L is és a benne található listák is fordított sorrendben vannak. Pl.:
[[1,1,2],[3,4]]--->[[4,3],[2,1,1]]