Tartalomjegyzék

Feladatok

Listák

1. *n* hosszú növekv? részek

Írjunk függvényt, mely kap egy listát és egy egész számot (n). Megkeresi az összes olyan n hosszú részintervallumot, amire igaz, hogy az elemei növekv? sorrendben vannak. Ezeket a listákat beteszi egy f? listába és ezt adja vissza. Segítség: bontsuk részfeladatokra!

2. Pascal

A Pascal-háromszög a binomiális együtthatók háromszög formában való elrendezése. Részletes leírás található pl. a magyar wikipédián. A lényeg az, hogy az *n.* sor *k.* eleme az az *"http://wiki.math.bme.hun alatt a k"http://wiki.math.bme.hu* binomiális együttható, és minden elem a felette lev? kett? összege. Írjuk meg a `pascal` nev? függvényt, ami visszaadja a Pascal-háromszög els? néhány sorát listák listájaként. A függvény paramétere:

- *n*, hogy hány sort számoljunk ki

Így tehát pl. `pascal(4)`-nek a következ?t kell visszaadnia:

```
[[1],
[1, 1],
[1, 2, 1],
[1, 3, 3, 1]]
```

3. Kiejtés

A föld bizonyos nyelvei a magyarok számára elég nehezen beszélhet?ek, például azért, mert az ? szavaikban sokkal több a mássalhangzó mint a mienkben. Magyarok számára például nehéz lehet kiejteni a szlovák zmrzlina szót, ami magyarul azt jelenti hogy fagylalt.

Írjunk python függvényt, ami megpróbálja megállapítani hogy egy szót nehéz-e kiejteni, a benne lev? mássalhangzók száma alapján.

A függvény neve legyen kiejtes, egy paramétere legyen: szo, egy idegen nyelv? szó. Az egyszer?ség kedvéért az itt leírt változat csak angol kis bet?ket fog tartalmazni. A függvény True-val térjen vissza, ha legalább kétszer annyi mássalhangzó van a szóban mint magánhangzó, és False-al egyébként. A lehetséges bet?k közül az a, e, i, o, u számít magánhangzónak.

4. Kicserélés

Írjunk egy python függvényt, ami 2 paraméterrel m?ködik: (szo,betu_1,betu_2). A függvény kap 1 stringet (szo), majd minden betu_1 bet?t kicserél benne betu_2 bet?re. Például f('ideges','e','a') -> 'idagas'

Szótárak

5. Leltárazás

A következ? probléma merül fel a boltokban: Egy leltáros felírja egy listában szépen sorjában, hogy mit lát maga el?tt. Ezekb?l a termékekb?l egy többször is szerepelni fog, attól függ?en, hogy hány darab van bel?le. Feladat: Írjunk egy függvényt, aminek a bemenete egy lista, a kimenete pedig egy szótár, aminek a kulcsai a listában szerepl? nevek, a kulcsokhoz pedig az ? darabszámuk tartozik.

6. Prímszám-e

Készítsünk egy szótárat, melyben a prímszámok vannak, mint kulcsok 2-t?l 100-ig, az értékek pedig igaz-hamis értékek, hogy az adott prím Mersenne-prím-e.
Ehhez 2 segéd függvényt érdemes írni:

- Visszaadja 2-t?l n-ig a prímek listáját (ezt már volt korábban)
- Megmondja egy prímszámról, hogy Mersenne prím-e: adjunk a számhoz 1-et, majd azt, hogy 2 hatvány-e tesztelhetjük úgy, hogy vizsgáljuk milyen maradékot ad 2-vel osztva, ha 0-t, osztjuk 2-vel, és vizsgáljuk tovább.

7. Ötösök

Miután kijavítottuk a vizsgát, és megvannak a százalékos eredmények, szeretnénk kisz?rni bel?le az ötösök listáját, hogy megdicsérhessük ?ket az el?adáson.

Írjunk egy python függvényt, ami ki tudja sz?rni az eredményekb?l az ötösöket. A függvény neve legyen `otosok`, és egy paramétere legyen

- *szazalekok*, egy python szótár (dict), ami minden NEPTUN-kódhoz tartalmazza az adott hallgató százalékos eredményét.

A függvény egy listát adjon vissza, amiben az ötöst elért hallgatók NEPTUN-kódjai vannak. Akkor ötös egy hallgató, ha legalább 85 százalékot elért.

8. Gólkirály

Koppány és barátai minden hétvégén játszanak egy barátságos focimeccset az egyik helyi focipályán. Szeretnék tudni az év végén hogy ki rúgta közülük a legtöbb gólt, hogy egy kicsit megünnepelhessék az illet? teljesítményét. Koppány megkért minket hogy írjunk egy python függvényt, ami segít a meccs végén elkönyvelni valaki góljait.

A függvény neve legyen `golkiraly`, és három paramétere legyen

- *eredmenyek*, az év folyamán eddig l?tt gólok száma, szótár formájában, amiben mindenkinek a nevéhez hozzá van rendelve hogy eddig mennyit l?tt
- *jatekos* és *darab* az elkönyvelend? gólok löv?jének neve, és a góljainak száma

A függvény adja vissza az *eredmenyek*-nek megfelel?en módosított változatát, azaz ha eddig nem szerepelt benne játékos, akkor most szerepeljen benne *darab*-bal, ha eddig is szerepelt benne, akkor pedig növeljük meg a hozzá könyvelt gólok számát *darab*-bal.