Tartalomjegyzék

Feladatok

Bevásárlás

A feladat a már elõadáson látott feladat megoldása. Azaz, írjunk olyan függvényt, melynek két bemenete két szótár. Az elsõ az árak szótár, mely a boltban található összes áruhoz hozzárendeli annak árát. A másik a mennyiségek szótár, mely kulcsként tartalmazza, hogy mit vettünk értékként, pedig hogy mennyit vettünk az adott termékbõl. Pl:

```
arak = {
'alma': 150,
'szilva': 190,
'ananász': 450,
'banán': 300}

mennyisegek = {
'banán': 0.6,
'alma': 1.5,
'ananász': 2 }

arak = {
 'alma': 150,
 'szilva': 190,
 'ananasz': 450,
 'banan': 300}

mennyisegek = {
 'banan': 0.6,
 'alma': 1.5,
 'ananasz': 2 }

def bevasarlas(ar, menny):
    s = 0
    for aru in menny:
        s += ar[aru] * menny[aru]
    return s
```

```
print bevasarlas(arak, mennyisegek)
```

CloudCoder

osztalyzas

```
def osztalyzas(szazalekok, neptunkod):
    pont = szazalekok[neptunkod]
    if 85 <= pont:
        return 5
    if 70 <= pont:
        return 4
    if 65 <= pont:
        return 3
    if 40 <= pont:
        return 2
    return 1
```

vagy

```
def osztalyzas(szazalekok, neptunkod):
    return max(0, szazalekok[neptunkod] - 25) / 15 + 1
```

otosok

```
def otosok(szazalekok):
    uj = []
    for neptun in szazalekok:
        if szazalekok[neptun] >= 85:
            uj.append(neptun)
    return uj
```

golkiraly

```
def golkiraly(eredmenyek, jatekos, darab):
    if jatekos in eredmenyek:
        eredmenyek[jatekos] = eredmenyek[jatekos] + darab
    else:
        eredmenyek[jatekos] = darab
    return eredmenyek
```

enciklopedia

```
def enciklopedia(szocikkek):
    szocikkek.sort()
    d = {}
    for szo in szocikkek:
        if szo[0] not in d:
            d[szo[0]] = szo
    return d
```

leltar_frissites

```
def leltar_frissites(leltar, ujszam):
    for i in range(len(leltar)):
        if leltar[i] in ujszam:
            leltar[i] = ujszam[leltar[i]]
    return leltar
```

kozel

```
def skalar_szorzat(a, b):
    s = 0
    for i in range(len(a)):
        s += a[i] * b[i]
    return s

def kozel(l):
    nagy = skalar_szorzat(l[0], l[1])
    for i in range(len(l)):
        for j in range(len(l)):
            if i != j and nagy < skalar_szorzat(l[i], l[j]):
                nagy = skalar_szorzat(l[i], l[j])
    return nagy
```