Tartalomjegyzék

- 1 6. gyakorlat - Referenciak, Hibakezelés
  - 1.1 Feladatok
    - 1.1.1 Házi feladat
  - 1.2 Feladatok megoldása
    - 1.2.1 lakoma_1
    - 1.2.2 lakoma_2
    - 1.2.3 lakoma_3
    - 1.2.4 lakoma_4
    - 1.2.5 nyeses
  - 1.3 Házi feladat megoldása
    - 1.3.1 nokedli
    - 1.3.2 nokedli_hiba

6. gyakorlat - Referenciak, Hibakezelés

Ezen a gyakorlaton az el?z? heti el?adás anyagát dolgozzuk fel.

Feladatok

Feladatok a CloudCoder-en megtalálhatóak. Ajánlott sorrend:

1. lakoma_1
2. lakoma_2
3. lakoma_3
4. lakoma_4
5. nyeses

Az els? négy feladat egymásra épül, ugyanazt a **lakoma** nev? függvényt fejlesztitek egyre tovább. Ezért amikor kész az egyik feladat, érdemes átmásolni az addigi megoldásotokat a következ? feladathoz. De vigyázzatok, az ellen?rz? kódot ne másoljátok át, csak a megoldásotokat!

A mostani feladatok mind tartalmaznak a kódvázlatban ellen?rz? kódot. Ehhez ne nyúljatok hozzá (a házi feladatnál pl. ha hozzányúltok, az hibásnak számít). Az se fontos, hogy megértsétek ami oda van írva, nem kell elolvasnotok. Ha véletlenül átírnátok, itt az eredeti kódvázlat mindegyik feladathoz:

lakoma_1:

```
def lakoma(vendegek, valtozasok):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

def lakoma_1(vendegek, valtozasok):
    assert lakoma(vendegek, valtozasok) == None
    return vendegek
```

lakoma_2:

```
def lakoma(vendegek, valtozasok):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

def lakoma_2(vendegek, valtozasok):
    ret = lakoma(vendegek, valtozasok)
    if ret != None:
        return "" + ret
    else:
        return vendegek
```

lakoma_3:

```
def lakoma(vendegek, valtozasok):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

def lakoma_3(vendegek, valtozasok):
    ret = lakoma(vendegek, valtozasok)
    if ret != None:
        return "" + ret
    else:
        return vendegek
```

lakoma_4:

```
def lakoma(vendegek, valtozasok):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

def lakoma_4(vendegek, valtozasok):
    try:
        ret = lakoma(vendegek, valtozasok)
    except Exception:
        return "Ellenor: hiba " +  sys.exc_info()[0].__name__
    if ret != None:
        return "" + ret
    else:
        return vendegek
```

nyeses:

```
def nyes(fa):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

def nyeses(fa):
    assert nyes(fa) == None
    return fa
```

Házi feladat

nokedli_hiba:

```
def eves(kupacok, hol, k):
    pass #TODO

def nokedli(kupacok, evesek):
    pass #TODO

################################
# Ellenorzo kod, ne modositsuk!

import datetime
def nokedli_hiba(kupacok, evesek):
    try:
        return nokedli(kupacok, evesek)
    except:
        return "Ellenor: mas hiba"
```

Feladatok megoldása

lakoma_1

```
def lakoma(vendegek, valtozasok):
    for nev, szam in valtozasok:
        vendegek[nev] += szam
```

lakoma_2

```
def lakoma(vendegek, valtozasok):
    for nev, szam in valtozasok:
        vendegek[nev] += szam
        if vendegek[nev] < 0:
            return "Hiba: Negativ vendegszam"
```

lakoma_3

```
def lakoma(vendegek, valtozasok):
    try:
        for nev, szam in valtozasok:
            vendegek[nev] += szam
            if vendegek[nev] < 0:
                return "Hiba: Negativ vendegszam"
    except:
        return "Hiba: Rossz bemenet"
```

lakoma_4

```
def lakoma(vendegek, valtozasok):
    try:
        for nev, szam in valtozasok:
            vendegek[nev] += szam
            if vendegek[nev] < 0:
                return "Hiba: Negativ vendegszam"
    except KeyError:
        return "Hiba: Rossz nev"
```

nyeses

```
def nyes(fa):
    i = 0;
    while i < len(fa):
        if fa[i] == []:
            del fa[i]
        else:
            nyes(fa[i])
            i += 1
```

Itt figyelni kell, hogy csak akkor van *i += 1*, amikor nem töröltem az elemet. A másik esetben, ha törlöm *fa[i]*-t, akkor ami eddig *fa[i+1]* volt, az fog a *fa[i]* helyre csúszni, úgyhogy ha nem akarom kihagyni, nem szabad *i*-t növelni.

Házi feladat megoldása

nokedli

```
def eves(kupacok, hol, k):
    kupacok[hol] = kupacok[hol] % k

def nokedli(kupacok, evesek):
    for hol, k in evesek:
        eves(kupacok, hol, k)

    return sum(kupacok)
```

nokedli_hiba

```
def eves(kupacok, hol, k):
    kupacok[hol] = kupacok[hol] % k

def nokedli(kupacok, evesek):
    try:
        for hol, k in evesek:
            eves(kupacok, hol, k)
    except IndexError:
        return "Hiba: Rossz hely"

    return sum(kupacok)
```