

# Szerencseegyenleg - A megoldás magyarázata

A megoldás a `luck_balance.py` szkript fájlban található, azon belül is a `luckBalance` függvény szolgáltatja a megoldást.

Elsőként rendezzük az előversenyeket a hozzájuk társított szerencseértékek szerint csökkenő sorrendbe, ugyanis minél nagyobb egy előversenyhez társított szerencseérték, annál inkább szeretné Léna elveszíteni, ezzel is növelve a szerencseegyenlegének értékét (vö. mohó stratégia). Ehhez meghívjuk a `contests` lista `sort` nevű tagfüggvényét. A tagfüggvény releváns paramétereinek megadásával beállítjuk, hogy a rendezés a társított szerencseértékek szerint történjen (vö. `key=lambda x: int(x[0])`), illetve, hogy csökkenő sorrendet várunk el (vö. `reverse=True`). A teljes vonatkozó kódsor tehát az alábbiak szerint alakul.

```
# rendezzuk a versenyeket a hozzájuk tartozó szerencseertekek szerint
# csokkeno sorrendbe (vö. mohó stratégia)
contests.sort(key=lambda x: int(x[0]), reverse=True)
```

Ezután beállítjuk Léna kezdő szerencseértékét:

```
# beallitjuk Lena kezdo szerencseegyenleget
luck = 0
```

Ezt követően bejárjuk a már rendezett `contests` lista elemeit (vö. `for c in contests`). Ha az aktuálisan vizsgált előverseny nem fontos (vö. `c[1] == 0`), akkor Léna nyugodtan elveszítheti ezzel is növelve szerencseegyenlegének értékét (vö. `luck += c[0]`). Ha azonban az előverseny fontos (vö. `c[1] == 1`), akkor a kérdés, hogy Léna el tudja-e veszíteni még (vö. `k > 0`). Ha igen, akkor ezzel is növeli szerencseegyenlegének értékét (vö. `luck += c[0]`), ugyanakkor ezzel párhuzamosan bukási lehetőségeinek száma csökken (vö. `k -= 1`). Ha viszont már nem tud több fontos versenyt elveszíteni, akkor szerencseegyenlegének értéke a versenyhez társított szerencseértékkel csökken (vö. `luck -= c[0]`). Az alábbiakban a teljes vonatkozó kódrészlet látható.

```
# végigmegyünk a versenyek tombjen
for c in contests:
    # ha a verseny nem fontos, akkor Lena nyugodtan elveszitheti
    if c[1] == 0:
        # ekkor no Lena szerencseegyenlege a versenyhez
        # tartozó szerencseertekekkel
        luck += c[0]
        continue
    # ha a verseny fontos, viszont Lenának meg van bukasi lehetosege
    if k > 0:
        # ekkor is no Lena szerencseegyenlege a versenyhez
        # tartozó szerencseertekekkel
        luck += c[0]
```

```
# viszont csokken Lena bukasi lehetosegeinek szama
k -= 1
continue
# ha a verseny fontos, viszont Lenanak mar nincs több bukasi
lehetosege,
# akkor Lena szerencseegyenlege csokken a versenyhez tartozo
# szerencseertekkel
luck -= c[0]
```

Végül miután minden előversenyt feldolgoztunk, visszaadjuk Léna szerencseegyenlegének aktuális értékét az alábbiak szerint:

```
# visszaadjuk Lena szerencseegyenlegének erteket a versenyek utan
return luck
```