

jacht (meg város) finder 2002

Készítették: Dobrovenszki Balázs, Szabó István – 10.c

Dávid (*fiktív személy*) nagyon gazdag. Dávid szabadidejében ellátogatott a világ legnagyobb városaiba és rengeteg vitorlás jachton utazott, Dávid ezeknek a helyeknek és jachtoknak leírta a legjellemzőbb adatait és személyes értékelését egy kiskönyvbe. Ezt a kiskönyvet majd odaadta nekünk azzal a kéréssel, hogy tudnánk-e neki elemezni az adatokat, mert már annyit mindent leírt, hogy egyszerűen nem tudja átlátni.

Ebben a programban Dávidnak a kiskönyvi adatait jelenítettük meg különböző szempontok szerint. A program fő aspektusa a felhasználóbarát, könnyen kezelhető felhasználói felület.

Az adatokat tartalmazó fájlok az alábbi módon vannak felépítve:

```
Ha1 neve;Súly (kg);Hossz (cm)
Harcsa;50;38.5
```

**A kép csak példa, nem Dávid kiskönyvének a konkrét adatai*

Az adattömb első sorában található meg az adat típusainak az elnevezése pontosvesszővel (";") elválasztva, ez gyakorlatilag egy fejlécként szolgál; ezután soronként található meg a különböző adatok, amelyek az alábbi fejlécekhez tartoznak, szintén pontosvesszővel elválasztva.

Két különböző fájlt készítettünk, ami tartalmazza a kiskönyv adatait:

1. be1.txt (Dávid utazott jachtjai: 4 különböző adattag; 457 különböző adat)
2. be2.txt (Nagyvárosok ahol Dávid járt: 5 különböző adattag; 78 különböző adat)

Ezekkel az adatokkal használhatja alaphoz a felhasználó a programot, de akár egy saját adattömbbel is működtetheti, feltéve hogy ez az adattömb követi a fentebb látható formát. Az adattagok száma nincs korlátozva az 5 adattaghoz, a program bármennyivel működik, amíg 2 vagy 2-nél több adattag van és csak a legelső adattag szöveg típusú, többi pedig egész vagy lebegőpontos szám.

A program indítása után a kezdőképernyő fogadja a felhasználót:

Lehetséges módok:

```
1.: Adat megjelenítő eszközök
2.: Adat rögzítés manuálisan
3.: Betűszín megváltoztatása
q.: Kilépés
```

Válaszd ki milyen módban szeretnéd elindítani a programot:

Itt a felhasználó 3 különböző opciót lát (*fenti kép*). Miután a felhasználó kiválasztott egy olyan módot, ami az adat fájlokkal kapcsolatos, a program elkéri az adott fájl nevét. **Az első módban**

a tanult programozási tételek alapján lehet megjeleníteni a meglévő adatokat a konzolon, vagy fájlba kiírni ("ki.txt"). A **második módban** a felhasználó a már meglévő adatokhoz tud hozzáfűzni saját adatot. A **harmadik módban** a konzol betűszínét lehet változtatni. A módok közötti lépegetést a visszalépés funkció biztosítja, amit a "q" betű lenyomásával érhetünk el.

Mivel a program sok szöveget használ ezért létrehoztunk egy külön fájlt, ami tárolja a szöveget és csináltunk rá egy függvényt, hogy tudjuk olvasni a fájl tartalmát majd megjeleníteni a felhasználó számára.

Függvény elemzés: Rendezés

```
329 def insertion_sort(L, ascending=False, wantindexes=False):
330     y = []
331     y_i = []
332     for i in range(len(L)):
333         y.append(L[i])
334         y_i.append(i)
335
336     for i in range(len(y)):
337         j = minimum_or_maximum(y, ascending, i)
338         if y[i] != y[j]:
339             y[i], y[j] = y[j], y[i]
340             y_i[i], y_i[j] = y_i[j], y_i[i]
341     if not wantindexes:
342         return y
343     else:
344         return y_i
```

A fent látható függvény a minimum/maximum kiválasztásos rendezés továbbfejlesztett verziója.

Paraméterek:

- **L** - a rendezendő lista
- **ascending** - ha igaz, a lista növekvő sorrendbe lesz rendezve, alapértelmezett érték: hamis
- **wantindexes** - ha igaz, a rendezett lista indexeit kapjuk vissza, alapértelmezett érték: hamis

A függvény először 2 különböző listát tölt fel, az egyikben a rendezendő lista értékeit, a másikban a lista indexeit tárolja el. Majd, a *minimum_or_maximum* függvény megkapja paraméterként a listát, a kezdő indexet és a fent említett *ascending* változót, ami eldönti, hogy a maximum vagy a minimum kiválasztás tételét alkalmazza. A listák elemeit megcseréli, ha rossz sorrendben vannak, majd a *wantindexes* változó alapján a megfelelő listát adja vissza.

**A függvény nem helyben rendez, ezért a visszatérési értéke mindig egy lista lesz.*

Példa a függvény használatára:

```
342         assorted = insertion_sort(output[3], chc == "2", True)
```

Itt az adatokat rendezi, ha a felhasználó 1-est ír be, akkor csökkenő sorrendbe, ha 2-est akkor növekvő sorrendbe.