resitev

January 28, 2024

Ana in Berta se neizprosno spopadata na dražbi. Izklicnih cen ni. Dražba poteka tako, da najprej da ponudbo Ana (ker je starejša), nato Berta, pa spet Ana, … dokler ena od njiju ne odneha.

In še eno (za reševalce dobrodošlo pravilo): Ana mora *vedno* dati ponudbo. (Pravilo si je izmislil lastnik dražbe, ki ima rad tarok in ga jezi, da *obvezna 3* v resnici ni obvezna, temveč le povzroča klope, ki jih osebno ne mara, ker ima takrat vedno najboljše karte.)

Dogajanje je popisano v datoteki takšne oblike.

```
11
17
24
30
-1
13
27
33
-1
12
27
34
40
-1
9
-1
8
20
30
31
-1
```

Tule se je prodajalo pet predmetov.

- Ana je za prvi predmet ponudila 11, Berta 17, Ana 25, Berta 30. Številka -1 označuje, da je predmet prodan. Prvega je torej kupila **Berta za 30**.
- Za drugi predmet je Ana ponudila 13, Berta 27. Prodano **Ani za 33**.
- Za tretjega Ana 12, Berta 27, Ana 34, Berta 40. Prodano **Berti za 34**.
- Za četrtega Ana 9. Prodano **Ani za 9**. (Šlo je samo za star polomljen dežnik, ki pa je imel za Ano neko emocionalno vrednost, ker jo je spominjal na njeno teto po očetovi strani iz Lesničnega.)
- Za petega Ana 8, Berta 20, Ana 30, Berta 31. **Berti za 31**.

Vidimo torej:

1. Skupno so prodali 5 predmetov.
2. Najvišja dosežena cena je 40.
3. Skupna cena prodanih predmetov je 143 (to je, 30 + 33 + 34 + 9 + 31).
4. Ana je kupila 2 predmeta, Berta pa 3.
5. Ana je porabila 42, Berta 101.

0.1 Obvezna naloga

Napiši program, ki prebere datoteko in izračuna ter izpiše dejstva do točke 3.

0.2 Dodatna, neobvezna naloga

Dodaj še točki 4 in 5.

0.3 Podatki

Podatki so v datotekah drazba.txt (gornji primer) in drazba-dolga.txt, pri kateri je rezultat

```
Prodanih predmetov: 42
Najvišja cena: 266
Skupna cena: 4736
Ana: 19 predmetov za 2217
Berta: 23 predmetov za 2519
```

Da bo rešitev priznana za pravilno, mora izpisati pravilne rezultate (besedilo ni pomembno, pomembne so številke) za obe datoteki.

1 Rešitev

Če bomo reševali naloge v ločenih zankah (in ne vseh v eni sami veliki), bomo v vsaki, vedno znova, klicali funkcijo `open`. Da ne bo potrebno vedno spreminjati `open("drazba.txt")` v `open("drazba-dolga.txt")` najprej definirajmo spremenljivko `ime_dat` z imenom datoteke, potem pa vedno kličimo kar `open(ime_dat)`.

[2]:
```
ime_dat = "drazba.txt"
```

1.0.1 Število prodanih predmetov

Da izvemo število prodanih predmetov, preprosto preštejemo, kolikokrat se v datoteki pojavi "cena" `-1`.

[4]:
```
predmetov = 0
for cena in open(ime_dat):
    if int(cena) == -1:
        predmetov += 1
print("Prodanih predmetov:", predmetov)
```

```
Prodanih predmetov: 5
```

Lahko bi pisali tudi

```
for vrstica in open(ime_dat):
    cena = int(vrstica)
    if cena == -1:
    ...
```

vendar ceno tako ali tako uporabimo le enkrat, torej je vseeno, če jo pretvorimo v število kar znotraj `if`-a.

Morda se je komu zahotelo pisati

```
for vrstica in open(ime_dat):
    if vrstica == -1:
```

to ne deluje, ker je vrstica niz in ne število, torej ni enaka (`int`-u) `-1` temveč nizu `"-1"`. To ni isto.

Ako bi dotični odkril, kaj je narobe bi to morda hotel popraviti v

```
for vrstica in open(ime_dat):
    if vrstica == "-1":
```

tudi to ne deluje, ker je na koncu vrstice še znak za konec vrstice `\n`, ki smo ga (ali pa tudi ne) omenili na predavanju, omenjen pa je v zadnji nalogi za vaje.

Delovalo bi

[5]:
```
predmetov = 0
for cena in open(ime_dat):
    if cena == "-1\n":
        predmetov += 1
print("Prodanih predmetov:", predmetov)
```

```
Prodanih predmetov: 5
```

ali

[6]:
```
predmetov = 0
for cena in open(ime_dat):
    if cena.strip() == "-1":
        predmetov += 1
print("Prodanih predmetov:", predmetov)
```

```
Prodanih predmetov: 5
```

To pa ni že nič preprosteje od `int`. Pravzaprav je slabše kot `int`. Vrstice datoteke vsebujejo števila, torej jih pretvarjajmo v števila.

1.0.2 Najvišja dosežena cena

Ta naloga je preprostejša, kot se je morda komu zdelo. Ko beremo cene, nam je vseeno, kdo je reč kupil in, sploh, ali je določena pravkar prebrana cena zadnja ali ne. Da rešimo nalogo, je dovolj, da poiščemo najvišje število v datoteki. To pa je naloga, ki smo jo reševali na predavanju (najvišja napovedana temperatura v Radovljici) in na vajah.

Po zgledu prejšnje rešitve lahko napišemo

[7]:
```
najvisja = 0
for cena in open(ime_dat):
    if int(cena) > najvisja:
        najvisja = int(cena)
print("Najvišja cena:", najvisja)
```

```
Najvišja cena: 40
```

Ker tule dvakrat pokličemo `int(cena)`, pa se počasi spodobi, da v število pretvorjeno vrstico shranimo v novo spremenljivko. Lepše je torej

[8]:
```
najvisja = 0
for vrstica in open(ime_dat):
    cena = int(vrstica)
    if cena > najvisja:
        najvisja = cena
print("Najvišja cena:", najvisja)
```

```
Najvišja cena: 40
```

1.0.3 Skupna cena prodanega

Tu pa se stvari zapletejo. Potrebno je sešteti vse cene, ki se pojavijo *pred* ceno `-1`. Torej: ko naletimo na ceno `-1`, moramo k skupni vrednosti prodanih izdelkov prišteti ceno, ki se je pojavila predtem. Tega (za zdaj) ne znamo storiti drugače, kot da si jo zapomnimo.

[9]:
```
prej = 0
skupno = 0
for cena in open(ime_dat):
    cena = int(cena)
    if cena == -1:
        skupno += prej
    prej = cena
print("Skupna cena:", skupno)
```

```
Skupna cena: 143
```

Trik je na koncu zanke. Zadnja stvar, ki jo naredimo, je `prej = cena`. Na ta način bo trenutna `cena` v naslednjem krogu zanke dostopna v spremenljivko `prej`.

1.0.4 Vse tri skupaj

Z vsemi tremi točkami lahko opravimo tudi v eni zanki.

[11]:
```
prodanih = 0
najvisja = 0
prej = 0
skupno = 0
```

```
for cena in open(ime_dat):
    cena = int(cena)
    if cena == -1:
        prodanih += 1
        if prej > najvisja:
            najvisja = prej
        skupno += prej
    prej = cena

print("Prodanih predmetov:", predmetov)
print("Najvišja cena:", najvisja)
print("Skupna cena:", skupno)
```

```
Prodanih predmetov: 5
Najvišja cena: 40
Skupna cena: 143
```

1.1 Delitev med Ano in Berto

Preostali točki rešimo kar v enem zamahu. Bistvo naloge ni v tem, da znamo šteti in seštevati, kot v prvih dveh točkah, temveč v tem, da vemo, kdo je tisti, ki je na koncu koncev kupil nek predmet. Eden od načinov - za nas trenutno morda najpreprostejši - je, da štejemo, koliko višanj cene je doživel nek predmet. Če jih je liho, ga je kupila Ana, sicer Berta.

[18]:
```
ana = berta = 0
ana_skupno = berta_skupno = 0
visanj = 0
prej = -1
for cena in open(ime_dat):
    cena = int(cena)
    if cena != -1:
        visanj += 1
    else:
        if visanj % 2 == 1:
            ana += 1
            ana_skupno += prej
        else:
            berta += 1
            berta_skupno += prej
        visanj = 0
    prej = cena

print("Ana:", ana, "predmetov za", ana_skupno)
print("Berta:", berta, "predmetov za", berta_skupno)
```

```
Ana: 2 predmetov za 42
Berta: 3 predmetov za 101
```

Tako kot prej imamo še vedno `cena` in `prej`. Nova spremenljivka je število `visanj`, ki je v začetku

`0`. Vsakič, ko preberemo ceno, ki ni `-1`, gre za višanje, torej k `visanj` prištejemo 1. Sicer pa je bil izdelek pravkar (no, v prejšnji vrstici) prodan. Če je število višanj liho, povečamo število predmetov in skupno ceno za Ano, sicer za Berto. V obeh primerih pa (ne spreglejmo!) nastavimo `visanj` nazaj na 0.

Komur je ta naloga na robu tistega, kar razume, naj bo pozoren predvsem na to, kje v strukturi teh `if`-ov in `else`-ov se pojavita vrstici `visanj = 0` in `prej = cena`. Prva je znotraj `else`-a, ki pokrije primer, ko je izdelek prodan. Tako "resetiramo" število višanj za naslednji izdelek. Druga, `prej = cena` je izven pogojev, vendar v zanki, saj se mora zgoditi vedno. (Glede na to, da na dražbi vedno prodamo vse izdelke, ker Ana *mora* predlagati ceno, nikoli ne bomo naleteli na dve `-1` zapored, tako da bi bilo vseeno, če bi bila vrstica `prej = cena` znotraj prvega `if`, torej pod `visanj += 1`. Vendar takšna rešitev zahteva malo dodatnega razmisleka, struktura programa je bolj zoprna, pridobimo pa (skoraj) ničesar, zato raje napišimo, kot smo napisali. Naloga programerja je napisati pravilen program, ne pa dokazovati zvitosti z nepotrebnimi časovnimi optimizacijami ali kratkostjo programa na račun razumljivosti.)

1.2 Čisto vse skupaj

Če želimo vse rešiti v eni zanki, imamo v gornjem programu pravzaprav že skoraj vse, kar potrebujemo. Skupno število in cena prodanih predmetov je pač vsota cen in predmetov, ki sta jih (ločeno) kupili Ana in Berta. Le še najvišjo ceno si moramo zapomniti.

[19]:
```
najvisja = 0
ana = berta = 0
ana_skupno = berta_skupno = 0
visanj = 0
prej = -1
for cena in open(ime_dat):
    cena = int(cena)
    if cena == -1:
        if prej > najvisja:
            najvisja = prej
        if visanj % 2 == 1:
            ana += 1
            ana_skupno += prej
        else:
            berta += 1
            berta_skupno += prej
        visanj = 0
    else:
        visanj += 1
    prej = cena

print("Prodanih predmetov:", ana + berta)
print("Najvišja cena:", najvisja)
print("Skupna cena:", ana_skupno + berta_skupno)
print("Ana:", ana, "predmetov za", ana_skupno)
print("Berta:", berta, "predmetov za", berta_skupno)
```

```
Prodanih predmetov: 5
Najvišja cena: 40
Skupna cena: 143
Ana: 2 predmetov za 42
Berta: 3 predmetov za 101
```

1.3 Daljša dražba

Če želite preveriti, ali vse deluje tudi za daljšo dražbo, se vrnite nekaj celic višje, spremenite `ime_dat` v `ime_dat = "drazba-dolga.txt"` in s Shift-Enter izvajajte celico za celico.