

resitev

January 28, 2024

1 Dražba brez anonimnosti

Na dražbi seveda nimajo tako nepopolnih zapisnikov, kot je namigovala prejšnja naloga. Ve se, katere predmete prodajajo. Na dražbe tudi ne hodita le Ana in Berta; Ani seveda ni treba vedno začeti višanja.

V resnici so zapisniki videti tako:

```
slika,Berta,31
slika,Ana,33
slika,Berta,35
slika,Fanči,37
slika,Ana,40
slika,Fanči,45
pozlačen dežnik,Ema,29
Meldrumove vaze,Greta,44
Meldrumove vaze,Ana,46
```

Vsaka vrstica vsebuje tri podatke, ločene z vejico. Prvi je prodajani predmet, drugi je ime osebe, ki viša ceno, in tretji so ponujene cene teh oseb. Pri branju si bomo (lahko) pomagali s tem, da vemo naslednje.

- Vsak predmet nastopa le v zaporednih vrsticah: ko je slika prodana (Fanči za 45), se v datoteki ne pojavi nikoli več.
- Nobeno ime se ne pojavi dvakrat zapored pri istem izdelku. Nihče ne viša lastne ponudbe.
- Cene za posamezni izdelek se vedno le višajo.

Pri reševanju ne smete predpostavljati, da na dražbi sodelujejo le osebe, ki jih vidite v datoteki in da se prodajajo le ti izdelki. Program mora biti splošen in delovati tudi za datoteko, ki jo bodo na dražbi pripravili jutri - z drugimi predmeti in osebami.

Konkretno, program mora znati obdelati tako “zapisnik.txt” kot “zapisnik-dan2.txt”. (Ne vznemirjaj se, če se pri drugem zapisniku spoli oseb ne bodo ujemali in bo Anton *kupila* to in ono.)

1.1 Obvezna naloga

1. Izpiši, kateri predmet je dosegel najvišjo ceno, kdo ga je kupil in za koliko.
2. Izpiši končne cene vseh predmetov (vsak predmet v drugi vrstici).
3. Izpiši, koliko ponudb je bil deležen vsak izmed predmetov (vsak v drugi vrstici).
4. Izpiši, za kateri predmet je bilo največ ponudb. Če si prvo mesto deli več izdelkov, izpiši enega od njih.

Če želiš pustiti med izpisi prazno vrstico, pokliči `print` brez argumentov - `print()`.

Izpis je lahko takšen:

Najdražji predmet je kip - za 107 ga je kupila Dani

```
slika - 45
pozlačen dežnik - 29
Meldrumove vaze - 78
skodelice - 83
kip - 107
čajnik - 15
srebrn jedilni servis - 63
perzijska preproga - 21
```

```
slika - 6
pozlačen dežnik - 1
Meldrumove vaze - 12
skodelice - 12
kip - 29
čajnik - 1
srebrn jedilni servis - 14
perzijska preproga - 2
```

Najbolj so se pulile za predmet kip

1.2 Rešitev

Prva ni raketna znanost. Beremo vrstice in vsako razbijemo na ime osebe, predmeta in ceno. Primerjamo cene in shranjujemo najvišjo ceno in pripadajoči predmet.

```
[1]: naj_cena = 0
    for vrstica in open("zapisnik.txt"):
        kaj, kdo, koliko = vrstica.split(",")
        if int(koliko) > naj_cena:
            naj_cena = int(koliko)
            naj_kdo = kdo
            naj_kaj = kaj

    print("Najdražji predmet je", naj_kaj, "- za", naj_cena, "ga je kupila", naj_kdo)
```

Najdražji predmet je kip - za 107 ga je kupila Dani

Drugo bomo ugnali s trikom: za vsak predmet shranjujemo vse cene. Ker so ključi v slovarju enkratni (prirejanje vrednosti obstoječemu ključu le spremeni njegovo vrednost), bo v slovarju opisana zadnja cena. Ker se cene le višajo in ker se vsak izdelek prodaja le enkrat, bomo tako imeli zapisano ravno najvišjo ceno. :)

```
[2]: cene = {}
for vrstica in open("zapisnik.txt"):
    kaj, kdo, koliko = vrstica.split(",")
    cene[kaj] = int(koliko)

for kaj, koliko in cene.items():
    print(kaj, "-", koliko)
```

```
slika - 45
pozlačen dežnik - 29
Meldrumove vaze - 78
skodelice - 83
kip - 107
čajnik - 15
srebrn jedilni servis - 63
perzijska preproga - 21
```

Tretja je klasično preštevanje. Vsakemu predmetu ustreza element slovarja. Če predmet vidimo prvič, ga dodamo v slovar; pripadajoča vrednost, število višanj bo 0. Ob vsaki pojavitvi predmeta povečamo število višanj.

```
[3]: visanj = {}
for vrstica in open("zapisnik.txt"):
    kaj, _, _ = vrstica.split(",")
    if kaj not in visanj:
        visanj[kaj] = 0
    visanj[kaj] += 1

for kaj, koliko in visanj.items():
    print(kaj, "-", koliko)
```

```
slika - 6
pozlačen dežnik - 1
Meldrumove vaze - 12
skodelice - 12
kip - 29
čajnik - 1
srebrn jedilni servis - 14
perzijska preproga - 2
```

Isti slovar potem uporabimo za rešitev zadnje obvezne naloge. Podobna je prvi, kjer smo iskali izdelek z najvišjo ponujeno ceno. Tam smo primerjali cene ter si zapomnili najvišje cene in pripadajoče predmete. Tu pa gremo čez slovar, ki pove število ponudb za posamezni izdelek in si zapomnimo največje število ponudb in pripadajoči predmet. Razlika je le v tem, da smo tam podatke brali z datoteke, tu pa jih dobivamo iz slovarja.

```
[5]: naj_zelena = None
for kaj, koliko in visanj.items():
    if naj_zelena is None or visanj[kaj] > visanj[naj_zelena]:
```

```
naj_zelena = kaj  
  
print("Najbolj so se pulile za predmet", naj_zelena)
```

Najbolj so se pulile za predmet kip

Nedvomno se bodo pojavile tudi takšne rešitve.

```
[12]: print("Najbolj so se pulili za predmet", max(visanj, key=visanj.get))
```

Najbolj so se pulili za predmet kip

To boste dobili na Stack Overflowu, ChatGPTju ali od prijatelja, ki pozna Python. Ne počnite tega. Na ta način se ne naučite ničesar. Razen, če razumete, da so metode Pythona prvorazredni objekti in kaj počne `max` s poimenskim argumentom `key`. Sicer pa uporaba takšnih rešitev, četudi jih izbrskate sami, ni nič bolj poučna, kot če vam nalogo reši kdo drug.

O takšnih rešitvah se bomo učili, vendar takrat, ko bomo pametnejši in jih bomo tudi razumeli. Dotlej pa za vajo raje pišimo zanke in pogoje, saj nam bodo prišli prav takrat, ko hitrejših rešitev ne bo.

1.3 Dodatna naloga

5. Za vsako osebo izpiši, koliko je porabila na dražbi.
6. Za vsak izdelek izpiši, za koliko je bila končna cena višja od prve.

Izpis bi lahko bil takšen:

Poraba po osebah:

Berta - 98
Cilka - 78
Dani - 107
Ema - 29
Fanči - 45
Greta - 63
Helga - 21

Dviganje cen:

slika - 14
pozlačen dežnik - 0
Meldrumove vaze - 34
skodelice - 33
kip - 77
čajnik - 0
srebrn jedilni servis - 36
perzijska preproga - 5

Nalogi bo preprosteje rešiti hkrati. Sestavili bomo tri slovarje. Vsi bodo imeli za ključne predmete.

- `prvic` bo za vsak izdelek vseboval njegovo prvo ceno,
- `zadnjic` bo vseboval zadnjo ceno,

- kupec bo vseboval končnega kupca.

```
[7]: prvic = {}  
zadnjic = {}  
kupec = {}  
  
for vrstica in open("zapisnik.txt"):  
    kaj, kdo, koliko = vrstica.split(",")  
    koliko = int(koliko)  
    if kaj not in prvic:  
        prvic[kaj] = koliko  
    zadnjic[kaj] = koliko  
    kupec[kaj] = kdo
```

Poglejmo, kako jih polnimo. V slovar `prvic` bomo dodali predmet, ki ga prvič vidimo. Preprosto preverimo, ali je že tam; če ga ni, ga dodamo, če je, pa nič. V slovar `zadnjic` zapišemo ceno vsakega predmeta, ki ga vidimo. Ko to storimo zadnjič, bomo zapisali zadnjo ceno. Podobno v slovar `kupec` vpisujemo vse ponudnike. Zadnji zapisani ponudnik je potem dejanski kupec. Preverimo, ali so slovarji videti smiselno.

```
[8]: prvic
```

```
[8]: {'slika': 31,  
      'pozlačen dežnik': 29,  
      'Meldrumove vaze': 44,  
      'skodelice': 50,  
      'kip': 30,  
      'čajnik': 15,  
      'srebrn jedilni servis': 27,  
      'perzijska preproga': 16}
```

```
[9]: zadnjic
```

```
[9]: {'slika': 45,  
      'pozlačen dežnik': 29,  
      'Meldrumove vaze': 78,  
      'skodelice': 83,  
      'kip': 107,  
      'čajnik': 15,  
      'srebrn jedilni servis': 63,  
      'perzijska preproga': 21}
```

```
[10]: kupec
```

```
[10]: {'slika': 'Fanči',  
      'pozlačen dežnik': 'Ema',  
      'Meldrumove vaze': 'Cilka',  
      'skodelice': 'Berta',
```

```
'kip': 'Dani',
'čajnik': 'Berta',
'srebrn jedilni servis': 'Greta',
'perzijska preproga': 'Helga'}
```

Zdaj lahko seštejemo porabo po osebah. Da ne bo vedno `if`, bomo uporabili `setdefault`. (Obstaja tudi boljša reč, `defaultdict`, vendar jo na predavanjih včasih omenimo že tu, včasih kasneje).

```
[17]: poraba = {}
      dvig = {}
      for kaj, kdo in kupec.items():
          poraba.setdefault(kdo, 0)
          poraba[kdo] += zadnjic[kaj]

      print()
      print("Poraba po osebah:")
      for oseba in poraba:
          print(oseba, "-", poraba[oseba])
```

Poraba po osebah:

```
Fanči - 45
Ema - 29
Cilka - 78
Berta - 98
Dani - 107
Greta - 63
Helga - 21
```

Za konec pa se še sprehodimo po slovarjih z začetnimi in končnimi cenami ter izpišemo razlike.

```
[19]: for izdelek in prvic:
      print(izdelek, "-", zadnjic[izdelek] - prvic[izdelek])
```

```
slika - 14
pozlačen dežnik - 0
Meldrumove vaze - 34
skodelice - 33
kip - 77
čajnik - 0
srebrn jedilni servis - 36
perzijska preproga - 5
```

Nepomemben estetski detajl: lahko bi pisali tudi

```
[20]: for izdelek, prva in prvic.items():
      print(izdelek, "-", zadnjic[izdelek] - prva)
```

```
slika - 14
```

pozlačen dežnik - 0
Meldrumove vaze - 34
skodelice - 33
kip - 77
čajnik - 0
srebrn jedilni servis - 36
perzijska preproga - 5

Tako bi eno ceno dobili “zastonj” brez indeksiranja. To imamo vedno radi, tu pa izjemoma nisem naredil tako, ker mi je všeč simetrija: `zadnjic[izdelek] - prvic[izdelek]` je lepše kot `zadnjic[izdelek] - prva`. Lepe, simetrične, urejene, sistematične programe je lažje brati in vsebujejo manj napak.

Tretja možnost bi bila

```
[21]: for (izdelek, prva), zadnja in zip(prvic.items(), zadnjic.values()):  
      print(izdelek, "-", zadnja - prva)
```

slika - 14
pozlačen dežnik - 0
Meldrumove vaze - 34
skodelice - 33
kip - 77
čajnik - 0
srebrn jedilni servis - 36
perzijska preproga - 5

Vendar (a) tega še ne znamo, (b) je grozno in čudno in (c) predpostavlja, da bo vrednosti v obeh slovarjih zložene v enakem vrstnem redu, kar je čudna in nevarna predpostavka.