

Programiranje 1 - 2. pisni izpit

16. 8. 2022

- Za reševanje je na voljo 60 minut.
- Dovoljena je le uporaba urejevalnika kode in ukazne lupine.
- Brskanje po internetu, učilnici, Tomu, ... **ni** dovoljeno.

1. NALOGA [10 T]

Maja ima rada zanimiva številska zaporedja, še najbolj pa ji je pri srcu Fibonaccijevo zaporedje. A ker zna veliko začetnih členov Fibonaccijevega zaporedja že na pamet, jo zanimajo tudi podobna zaporedja, kjer prva člena nista $F_0 = 0$ in $F_1 = 1$. Pomagajte ji napisati funkcijo `je_skoraj_fib(zap)`, ki vrne `True`, če člene zaporedja, predstavljenega s tabelo `zap`, lahko izračunamo po predpisu $F_n = F_{n-1} + F_{n-2}$. Pri tem naj velja, da sta prva člena zaporedja kar prva elementa tabele: $F_0 = \text{tab}[0]$ in $F_1 = \text{tab}[1]$.

Zgled:

```
>>> je_skoraj_fib([])
True
>>> je_skoraj_fib([42])
True
>>> je_skoraj_fib([33, 12])
True
>>> je_skoraj_fib([1, 1, 2, 3, 5, 8, 13, 21])
True
>>> je_skoraj_fib([0, 1, 1, 2, 3, 5, 8, 13])
True
>>> je_skoraj_fib([1, 0, 1, 2, 3, 5, 8, 13])
True
>>> je_skoraj_fib([3, 4, 7, 11, 18, 29])
True
>>> je_skoraj_fib([1, 2, 3, 4, 5])
False
>>> je_skoraj_fib([1, 1, 2, 3, 5, 8, 12])
False
>>> je_skoraj_fib([1, 1, 3, 4, 7, 11, 18])
False
```

2. NALOGA [12 T]

V popularni trgovini z oblačili X & Y morajo ob koncu obračunskega obdobja opraviti popis zalog. Ker so oblačila med seboj pomešana, zaposleni po vrsti v računalnik vnašajo pregledane artikle. Tako dobijo tabelo artiklov, kjer je posamezen element niz, ki predstavlja tip artikla in velikost, med seboj ločena s poševnico. Primer tabele, ki predstavlja zalogo: ["majica/s", "majica/m", "jopa/l", "majica/s"].

Ker se je iz takšne tabele težko znajti, vas prosijo za pomoč pri pisanju funkcije, ki bo podatke preuredila v slovar. Ključi slovarja naj bodo tipi artiklov, vrednosti pa slovar velikosti in ustrezno število artiklov. Napišite funkcijo `popis_zalog(artikli)`, ki sprejme tabelo artiklov in vrne slovar oblike `{"tip": {"velikost": st_ponovitev}}` kot kaže zgled.

Zgled:

```
>>> popis_zalog(['majica/s', 'majica/s', 'majica/s'])
{'majica': {'s': 3}}
>>> popis_zalog(['hlače/s', 'majica/m', 'jopa/l', 'jopa/s',
'hlače/m', 'hlače/s'])
{'hlače': {'s': 2, 'm': 1}, 'majica': {'m': 1},
'jopa': {'s': 1, 'l': 1}}
>>> popis_zalog([])
{}
```

3. NALOGA [13 T]

V Izgubljeni vasi organizirajo tekmovanje v orientacijskem teku. Za vsakega tekmovalca si v datoteko zapišejo, koliko minut po začetku tekmovanja je startal in koliko minut po začetku tekmovanja je priteknel v cilj. Sedaj morajo izračunati čase posameznih tekmovalcev in jih razvrstiti, da dobijo končni vrsti red.

Napišite funkcijo `izracunaj_case(vh_dat)`, ki iz vhodne datoteke `vh_dat` prebere vrstice oblike `<začetni_čas>-<končni_čas>`. Funkcija naj vrne tabelo parov zaporedne številke tekmovalca in njegovega časa, naraščajoče urejeno po časih.

Zgled:

Če imamo podano datoteko `cas1.txt`:

```
0 00-42
1 03-38
2 06-51
3 09-48
4 12-52
5 15-61
```

in pokličemo funkcijo z ukazom `izracunaj_case('cas1.txt')`, bo funkcija vrnila naslednjo tabelo:

```
[(2, 35), (4, 39), (5, 40), (1, 42), (3, 45), (6, 46)]
```

Torej, najboljši je bil drugi tekmovalec, ki je začel 3 minute po prvem startu in zaključil 38 minut po prvem startu - na poti je bil 35 minut. Na drugo mesto se je uvrstil 4. tekmovalec s časom 39 minut (09-48), tretji je bil 5. tekmovalec s časom 40 minut (12-52) in tako dalje.

4. NALOGA [15 T]

Mirko pri svojem programerskem projektu veliko računa z vektorji v dvodimenzionalnem prostoru oblike:

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Da bi si olajšal delo, želi napisati razred, ki bo poskrbel za izvajanje računskih operacij nad vektorji. Pomagajte mu:

- [3 T] Sestavite razred **Vektor**, ki sprejme komponenti x in y . Komponenti vektorja naj hrani v privatni spremenljivki `self._komponente` (kot tabelo, ki ima na prvem mestu komponento x , na drugem pa komponento y). Če konstruktorju podamo neveljavno vrednost (ni celo ali decimalno število), naj sproži napako `ValueError` kot kaže primer.
- [2 T] Razred naj vsebuje **lastnosti** x in y , ki vrmeta ustrezni komponenti. Lastnosti naj ne bosta spremenljivi.
- [2 T] Napišite metodi `__str__` in `__repr__`, ki vrmeta ustrezno predstavitev vektorja z nizom.
- [5 T] Sestavite ustrezni metodi za seštevanje in množenje vektorjev. Vektor lahko seštejemo le z drugimi vektorji, množimo pa ga lahko z vektorji (skalarni produkt) ter s celimi ali decimalnimi števili z leve ali z desne. V primeru neveljavnega operanda naj se sproži napaka `ValueError` z obvestilom "Napačen tip za seštevanje." oziroma "Napačen tip za množenje."
- [3 T] Napišite ukaze, ki s pomočjo napisanega razreda izračunajo in izpišejo vrednost izraza:

$$0.5 * \left(\begin{bmatrix} 3 \\ -1 \end{bmatrix} \cdot 3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 4 \end{bmatrix} \right) \quad (1)$$

Zgled:

Ustvarjanje objekta in lastnosti:

```
>>> v1 = Vektor(1, 3)
>>> v1.x
1
>>> v1.y
3
>>> Vektor(1, '2')
...
ValueError: Komponenti vektorja morata biti števili.
```

Izpis objekta:

```
>>> v2 = Vektor(1, 2)
>>> v2
Vektor(1, 2)
>>> print(v2)
[1, 2]
```

Seštevanje:

```
>>> v1 + v2
Vektor(2, 5)
>>> v1 + 42
...
ValueError: Napačen tip za seštevanje.
```

Množenje:

```
>>> v2 * 10
Vektor(10, 20)
>>> 0.5 * v2
Vektor(0.5, 1.0)
>>> v1 * v2
7
>>> v1 * "13"
ValueError: Napačen tip za množenje.
```