

Računalništvo (2. letnik fizike): 2. izpit

24. marec 2010

Čas reševanja: 120 minut

Reši vse naloge. Doseženih 100 točk šteje za maksimalno oceno.

Naloga 1 [25+10 točk]

Sestavi funkcijo `norm(imena)`, ki sprejme seznam imen, od katerih je vsako bodisi oblike "Ime Priimek" bodisi oblike "Priimek, Ime". Funkcija vrne nov seznam *normaliziranih* imen, to je imen oblike "Priimek, Ime". Primer uporabe:

```
>>> norm(["Janez Novak", "Depp, Johnny", "Dolly Parton"])
["Novak, Janez", "Depp, Johnny", "Parton, Dolly"]
```

Dodatna naloga [+10 točk]: Sestavi še funkcijo `uredi(imena)`, ki sprejme seznam imen in ga uredi po priimkih. Imena morajo v rezultatu nastopati v enaki obliki kot v vhodnem seznamu.

```
>>> imena = ["Janez Novak", "Depp, Johnny", "Dolly Parton"]
>>> uredi(imena)
>>> imena
["Depp, Johnny", "Janez Novak", "Dolly Parton"]
```

Naloga 2 [25 točk]

Dan je razred `Drevo`, ki predstavlja podatkovno drevo s poljubnim številom otrok v vsakem vozlišču:

```
1 class Drevo():
2     def __init__(self, vrednost, otroci=[]):
3         self.vrednost = vrednost
4         self.otroci = otroci
5
6     def __repr__(self):
7         # definirana v datoteki na ucilnici
```

Na primer, če imamo drevo s korenem 15, na katerega so pritrjeni listi 5, 7 in 9 ter poddrevo s korenem 11 ter otrokoma 12 in 13, to predstavimo takole:

```
>>> d = Drevo(15, [Drevo(5), Drevo(7), Drevo(9), Drevo(11, [Drevo(12), Drevo(13)])])
>>> d
+-----+
|      15      |
| 5 7 9 +-----+ |
|      | 11 | |
|      |12 13| |
|      +-----+ |
+-----+
```

Razredu `Drevo` dodaj metodo `spust(self)`, ki vrne seznam vozlišč na poti od `self` do najglobjega vozlišča v poddrevesu, ki se začne v `self`. Če je najgloblji vozlišč več, je vseeno, katerega metoda izbere. Primer za zgoraj opisano drevo:

```
>>> d.spust()
[15, 11, 12]
```

Veljaven rezultat bi bil tudi `[15, 11, 13]`.

Naloga 3 [25+10 točk]

Dan je seznam števil $a = [a_1, a_2, \dots, a_n]$, ki ga želimo razdeliti v dve skupini tako, da sta vsoti števil v skupinah enaki. Sestavi funkcijo `delitev(a)`, ki sprejme seznam števil in vrne seznam b dolžine n , sestavljen iz ničel in enic, ki tovrstno razdelitev opisuje takole: če je $b_i = 0$, dodelimo a_i v prvo skupino, če pa je $b_i = 1$, dodelimo a_i v drugo skupino. Vsoti števil v tako opisanih skupinah morata biti enaki. Če to ni možno, funkcija vrne `None`. Primer uporabe:

```
>>> delitev([5, 7, 13, 3, 4])
[0, 0, 1, 1, 0]
>>> delitev([1, 6, 1, 3, 12])
None
```

Kadar je možnih več rešitev, je vseeno, katero funkcija najde.

Namig, ki pa ga ni treba upoštevati: uporabi sestopanje. Sestavi funkcijo `delitev2(a, k, m)`, ki deluje podobno kot `delitev`, vendar vrne delitev seznamoma a na dva taka seznama, da je vsota prvega enaka k in vsota drugega m . Funkcijo `delitev` dobiš z uporabo funkcije `delitev2`.

Dodatna naloga [+10 točk]: Če rešitev ne obstaja, naj funkcija `delitev` namesto `None` vrne tisto razdelitev v dve skupini, pri kateri se vsoti skupin najmanj razlikujeta.

Naloga 4 [25 točk]

Lokalna telefonska številka je niz znakov, sestavljen iz števk, presledkov ter znakov `-` in `/`. Tvorijo jo tri skupine števk: krajevna koda, druga skupina in tretja skupina. Krajevna koda se začne z natanko eno ničlo in je sestavljena iz dveh ali treh števk, na primer 01 ali 031. Druga in tretja skupina sta sestavljeni iz treh ali štirih števk. Skupine so lahko ločene z znaki presledke, `-` in `/`; kateri so dovoljeni kje, razberi iz primerov:

Veljavno	Neveljavno
01 123 456	01 4776 666 123
01/1234 456	01 4776 666
041 6345-1234	001 1234 456
090 7123-456	05 12-3456
066/124-4569	41 6345-1234
031712456	066/ 124-4569
0711234567	066/1247-45
08912345678	066/12-4597
	041-6345-124
	066 /124 777

Mednarodna telefonska številka je sestavljena iz štirih skupin: mednarodna koda, na primer +386, krajevna koda brez začetne ničle, druga in tretja skupina. V mednarodni kodi nastopa ena do tri števke.

1. [10 točk] Sestavi funkcijo `telefonska(txt)`, ki vrne `True`, če podani niz predstavlja veljavno telefonsko številko, sicer `False`.
2. [5 točk] Sestavi funkcijo `globalno(txt)`, ki v besedilu `txt` vse lokalne telefonske številke spremeni v mednarodne tako, da odstrani vodilno ničlo in spredaj pripne +386. Telefonskih števk, ki so že v mednarodnem formatu, ne spreminja. Primer:

```
>>> globalno('''Pokličite nas na 090 123 345 ali našega predstavnika
v Bejrutu na +961 24 123 456, 24 ur na dan, 7 dni v tednu.'''')
"Pokličite nas na +386 90 123 345 ali našega predstavnika
v Bejrutu na +961 24 123 456, 24 ur na dan, 7 dni v tednu."
```

3. [10 točk] Sestavi funkcijo `globalno2(txt)`, ki v besedilu `txt` najde vse lokalne in mednarodne telefonske številke in jih *normalizira*, kar pomeni, da lokalne številke spremeni v mednarodne in vsem številkam (tudi mednarodnim) odstrani vse vmesne znake (presledke, `-` in `/`). Primer:

```
>>> globalno2('''Pokličite nas na 090 123-345 ali našega predstavnika
v Bejrutu na +961/24 123 456, 24 ur na dan, 7 dni v tednu.'''')
Pokličite nas na +38690123345 ali našega predstavnika
v Bejrutu na +96124123456, 24 ur na dan, 7 dni v tednu."
```