Izpit iz predmeta Programiranje 1, 1. februar 2011

Pri reševanju nalog je dovoljena raba standardnih Pythonovih knjižnic (`os`, `random`, `math`...). Ali je neka knjižnica standardna knjižnica ali ne, boste najlažje prepoznali po tem, ali je opisana v Pythonovi dokumentaciji.

Funkcije naj imajo takšna imena, kot jih predpisuje naloga. V rešitvah naj bo označeno, kateri del kode predstavlja rešitev katere naloge. Če naloga zahteva, naj funkcija *vrne rezultat*, mora *vrniti rezultat*, ne pa *izpisovati*. Funkcija mora podatke dobiti prek argumentov, kot določa naloga.

Rešitve vseh nalog shranite v eno samo datoteko .py in jo oddajte prek Učilnice tako, kot ste oddajali domače naloge.

Za testiranje lahko uporabljate testne primere, vendar se zavedajte, da je lahko program napačen tudi, če pravilno reši vse pripravljene testne primere.

Pri reševanju nalog je dovoljena vsa literatura na poljubnih medijih.

Študenti s predolgimi vratovi in podobnimi neželenimi lastnostmi bodo morali zapustili izpit, katerega opravljanje se bo štelo kot neuspešno. Hujše kršitve bomo prijavili disciplinski komisiji za študente.

Nalogi A in B se ne točkujeta, temveč sta obvezni za uspešno opravljen izpit. Kdor ju ne reši pravilno, je s tem že padel.

Ostalih šest nalog je vrednih enako število točk (tudi prva in zadnja!).

A. Praštevilski delitelji (obvezna naloga!)

Vzemi **SVOJO** rešitev domače naloge Razcep na prafaktorje. (Samo tisti, ki naloge niso oddali, oziroma jim je bila ocenjena negativno, naj vzamejo objavljeno rešitev.) Spremeni funkcijo `razcep` tako, da bo namesto razcepa na praštevila vrnila le seznam praštevil, ki delijo dano število. Torej: za domačo nalogo je morala funkcija delovati tako:

```
>>> razcep(252)
[(2, 2), (3, 2), (7, 1)]
>>> razcep(1944)
[(2, 3), (3, 5)]
```

Spremenite jo, da bo delovala tako:

```
>>> razcep(252)
[2, 3, 7]
>>> razcep(1944)
[2, 3]
```

B. Snežinke (obvezna naloga!)

Vzemi **SVOJO** rešitev domače naloge Snežinke. (Samo tisti, ki naloge niso oddali, oziroma jim je bila ocenjena negativno, naj vzamejo objavljeno rešitev.) Spremeni jo tako, da snežinka, ki pride do tal, ostane tam in ne pada več (namesto da bi se ponovno pojavila na vrhu).

1. Določanje spola

Številka EMŠO je sestavljena iz trinajst števk v obliki DDMMLLL50NNNX, pri čemer je DDMMLLL rojstni datum, 50 je koda registra, NNN je zaporedna številka in X kontrolna številka. Trimestna številka, NNN, je med 000 in 499 za moške ter 500-999 za ženske.

Napiši funkcijo **`jeZenska(emso)`**, ki za številko EMŠO, ki jo podamo ***kot niz***, vrne `True`, če pripada ženski in `False`, če moškemu.

Primer

```
>>> jeZenska('2206954500424')
False
>>> jeZenska('0502933501561')
False
>>> jeZenska('2002996506452')
True
>>> jeZenska('0802923508251')
True
```

2. Srečni gostje

Za ***okroglo mizo*** sedijo (okrogli?) gosti. Nekateri so srečni, drugi ne. Konkretno: srečni so moški, ki sedijo med dvema ženskama in ženske, ki sedijo med dvema moškima.

Razpored gostov podamo s seznamom njihovih številk EMŠO. Napiši funkcijo **`stevilo_srecnezev(razpored)`**, ki prejme takšen seznam in vrne število srečnih gostov.

Pazi: ker je miza okrogla, *sedi »prvi« gost poleg »zadnjega«*.

Če sedita moški in ženska sama za okroglo mizo, smemo predpostaviti, da sta srečna (navsezadnje ima v tem primeru ženska na vsaki strani enega (in istega) moškega in obratno).

Primer (ženske in moški so izpisani v primernih barvah; srečneži so podčrtani)

```
>>> stevilo_srecnezev(['0903912505707', '0110913506472', '2009956506012',
'1102946502619', '1902922506199', '2602930503913', '0204940508783', '1602960505003',
'0207959502025', '0207962509545'])
4
>>> stevilo_srecnezev(['1012947507186', '0506929507291', '3107910505475',
'1109984500497', '0510960506179', '0307978501042', '1607944505399', '1706954501918',
'1305934508423', '1406967504211'])
7
>>> stevilo_srecnezev(['0503973503512', '3004964501773', '1005933505567',
'2905936507573', '0712966507144'])
0
>>> stevilo_srecnezev(['0702948501362', '1505987508785'])
2
>>> stevilo_srecnezev(['2807955501835', '1604923501254', '0601925504908'])
0
>>> stevilo_srecnezev(['2807955501835'])
0
>>> stevilo_srecnezev([])
0
```

3. Gostoljubni gostitelji

Napiši funkcijo **`razporedi(gosti)`** za načrtovanje srečnih zabav. Funkcija dobi seznam gostov (spet kot seznam številk EMŠO) in mora vrniti seznam, ki je preurejen tako, da so gostje čim srečnejši. Pri tem se lahko zgodi, da je moških več kot žensk ali obratno, ali pa celo, da so na zabavi le moški ali le ženske.

(Da ne boste predolgo ugibali očitnega: idealni razpored dobimo, če postavimo moške v eno kolono, ženske v drugo, potem izmenično jemljemo iz vsake po enega in na koncu posedemo nesrečne pripadnike tistega spola, ki ostane.)

Primer (rešitve niso enolične – enako srečnost gostov je možno dobiti tudi z drugačnimi razporedi!)

```
>>> razporedi(['0505913509174', '2202973506004', '0304943506069', '2702943501809',
'2407980508463', '0209965503761', '2109913502875', '1802924506701', '0207970500808',
'1501917509568'])
['2702943501809', '0505913509174', '0209965503761', '2202973506004',
'2109913502875', '0304943506069', '0207970500808', '2407980508463', '1802924506701',
'1501917509568']
>>> razporedi(['2806984508656', '0602925509884', '1102979507594', '1104915509537',
'1502929506188', '1104923504226'])
['1104923504226', '2806984508656', '0602925509884', '1102979507594',
'1104915509537', '1502929506188']
>>> razporedi(['2806984508656', '0602925509884', '1102979507594', '1104915509537'])
['2806984508656', '0602925509884', '1102979507594', '1104915509537']
>>> razporedi(['2806984508656'])
['2806984508656']
>>> razporedi([])
[]
```

4. Presedanje

Neki dan so za okroglo mizo sedele Ana, Berta, Cilka, Dani in Ema (v tem vrstnem redu). Bile so nesrečne, zato so se drugi dan presedle v drug vrstni red: Cilka, Dani, Ema, Ana in Berta. Ni pomagalo saj so, če dobro razmislimo ... sedele enako, saj je miza okrogla so se pravzaprav le zavrtele – vsaka je imela še vedno isto levo in desno sosedo.

Napiši funkcijo **`enakaRazporeda(razpored1, razpored2)`**, ki pove, ali sta dva razporeda enaka ali ne. Razporeda sta podana s seznamom imen.

Če ti pride prav, smeš predpostaviti, da imajo vsi gostje različna imena (ali pa da seznam namesto imen vsebuje Enotne matične številke občank). Če ne razumeš, zakaj bi bilo to enostavneje, se ne vznemirjaj.

Primer

```
>>> r1 = ["Ana", "Berta", "Cilka", "Dani", "Ema"]
>>> enaka_razporeda(r1, ["Cilka", "Dani", "Ema", "Ana", "Berta"])
>>> True
>>> enaka_razporeda(r1, ["Berta", "Ana", "Cilka", "Dani", "Ema"])
False
>>> enaka_razporeda(r1, ["Ana", "Berta", "Cilka"])
False
>>> enaka_razporeda(r1, ["Greta", "Helga"])
False
>>> enaka_razporeda(r1, [])
False
>>>
>>> enaka_razporeda([], [])
True
```

5. Hosts

V Unixu (vključno z Linuxom in Mac OS x) se v imeniku **`/etc`** nahaja skrivnostna datoteka `hosts`. Na MS Windows jo najdemo v `c:\Windows\System32\drivers\etc`. V njej so shranjene pomembne reči: videti je lahko, recimo, takole

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost


212.235.188.24 google.com      # bolj zadane!
212.235.188.24   www.google.com        # bolj zadane!


193.2.72.47 bing.com
193.2.72.47   www.bing.com
```

Pravila za branje so takšna:

- če je v vrstici kakšen `#`, odbijemo vse od # do konca;
- pobrišemo ves beli prostor na začetku in koncu (`strip`);
- če ne ostane nič več, vrstico ignoriramo;
- sicer pa ostaneta v vrstici dve stvari: IP in neko ime.

Sestavite funkcijo **`read_hosts()`**, ki prebere datoteko `hosts` (predpostaviti smete, da se nahaja kar v trenutnem direktoriju) napisano po teh pravilih ter vrne slovar, katerega ključi so imena, vrednosti pa IPji. Zaradi preprostosti smete predpostaviti, da so vrstice pravilno oblikovane, torej, da vsebujejo natanko dve stvari, ločene s presledki in/ali tabulatorji.

(Koristen namig: na svojem domačem računalniku preverite datoteko `hosts` in če ne vsebuje teh štirih vrstic, ju dodajte, saj bo to izboljšalo delovanje iskalnikov.)

Primer

```
>>> read_hosts()
{'google.com': '212.235.188.24', 'bing.com': '193.2.72.47', 'www.bing.com':
'193.2.72.47', 'www.google.com': '212.235.188.24'}
```

6. URLji

URL je niz, ki se začne s `http://` ali `https://` in nadaljuje vse do prvega presledka, tabulatorja ali znaka za novo vrsto. Vsebuje lahko torej tudi poljubne znake, kot so oklepaji, pike, plusi in minusi ... torej karkoli razen belega prostora. Napišite funkcijo `find_URLs(s)`, ki kot argument prejme niz, kot rezultat pa vrne seznam vseh URLjev v njem.

Primer

```
s = """O regularnih izrazih lahko preberes v Pythonovi - http://www.python.org -
dokumentaciji, konkretno tule - http://docs.python.org/library/re.html - ampak ne med
izpitom. Zapiski o regularnih izrazih so na
http://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=5482 (deluje pa tudi
https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=5482)."""


>>> print find_URLs(s)
>>> ['http://www.python.org', 'http://docs.python.org/library/re.html',
'http://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=5482',
'https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=5482).']
```

Naj vas ne vznemirja, da bodo nekateri URLji (zadnji iz primera, recimo) nekoliko napačni, ker se jih bo na koncu držalo še kaj, kar ne sodi zraven.