Izpit iz predmeta Programiranje 1, 10. februar 2010 ob 11.00

Pri reševanju nalog je dovoljena domiselnost. Bližnjice do pravilne rešitve nalog so vredne pohvale. Dovoljena je raba standardnih Pythonovih knjižnic (`os`, `random`, `math`...), prepovedane pa so dodatne knjižnice, kot so PIL, numpy, PyQt... 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, ne prek zunanjih (npr. globalnih) spremenljivk, če ni eksplicitno določeno drugače.**

Vse naloge so vredne enako število točk. (Dodatne točke pri drugi nalogi so ... dodatne.)

Rešitve vseh nalog shranite v eno samo datoteko .py in jo oddajte prek tega strežnika na enak način, kot ste oddajali domače naloge. Uporabniki prenosnikov bodo rešitve oddajali na USB ključih.

Pri reševanju nalog je dovoljena vsa literatura na poljubnih medijih. Prepovedana pa je uporaba interneta, z izjemo učilnice (različice, do katere imate dostop s teh računalnikov) in vsi drugi načini komunikacije. Na računalniku mora biti izključena brezžična povezava in bluetooth ter ugasnjen Skype in drugi programi ter priprave, ki jih je mogoče uporabiti za komuniciranje. Kršilci teh pravil bodo zapustili izpit, katerega opravljanje se bo štelo kot neuspešno. Hujše kršitve bomo prijavili disciplinski komisiji za študente.

1. Stopnice

Stopnišča z neenakimi višinami stopnic lahko opišemo tako, da navedemo višine stopnic, merjene od tal (ne od prejšnje stopnice). Po stopnicah, opisanih s takšnim seznamom, pleza robot, ki lahko stopi največ 20 cm visoko. Napiši funkcijo `kakoVisoko(stopnice)`, ki kot argument prejme višine stopnic, rezultat, ki ga vrne, pa pove, kako visoko bo robot priplezal.

Primer

```
>>> print kakoVisoko([10, 20, 25, 45, 50, 71, 98, 110])
50
>>> print kakoVisoko([30, 40, 50])
0
>>> print kakoVisoko([10, 20, 30, 40, 45])
45
```

V prvem primeru robot pripleza do višine 50, nato pa se ustavi, ker je naslednja stopnica 21 cm višja od te, na kateri stoji. V drugem se ne more povzpeti niti na prvo stopnico. V zadnjem pripleza do vrha.

2. Eboran

Napiši funkcijo `eboran(stavek)`, ki prejme stavek (zapisan kot besede, med katerimi so presledki, brez ločil) in vrne nov stavek, v katerem so vse besede obrnjene.

Primer

```
>>> print eboran("vse je narobe tudi tale stavek")
esv ej eboran idut elat kevats
```

Pozor, "kevats elat idut eboran ej esv" ni pravilna rešitev naloge, vrstni red obrnjenih besed mora ostati enak.

Za dodatnih 10 % (npr. če želite pisati 110 %) lahko rešite nekoliko težjo nalogo, ki zna delati z ločili, večkratnimi presledki in vsem drugim, kar se lahko pojavlja znotraj stavka:

Primer

```
print eboran2('zapel je: "vse je narobe, tudi     tale stavek."')
lepaz ej: "esv ej eboran, idut     elat kevats."
```

3. Križanka

Napišite funkcijo `krizanka(beseda, besede)`, ki dobi kot argument besedo `beseda`, v kateri so manjkajoče črke zamenjane s pikami, in seznam besed `besede`. Funkcija naj vrne seznam vseh besed, ki se ujamejo s podano besedo.

Primer

```
>>> krizanka("r.k.", ["reka", "rokav", "robot", "roka"])
['reka', 'roka']
```

Namig: če ti je nalogo pretežko rešiti v eni funkciji, najprej pripravi funkcijo, ki dobi dve besedi in pove, ali druga ustreza prvi. To funkcijo nato kliči v funkciji, ki jo zahteva naloga.

4. Popularni rojstni dnevi

Napiši funkcijo `histogramDni(imedat)`, ki kot argument dobi ime datoteke, v kateri je seznam številk EMŠO, v vsaki vrstici po ena. Nato izračuna, koliko ljudi je rojenih na posamezni dan v letu in to izpiše (po dnevih). Izpis mora biti v obliki, kakršno vidite v primeru (vključno s presledki pred in med številkami). Dneve, na katere ni rojen nihče, naj izpusti. Vrstni red dni ni pomemben.

EMŠO je sestavljena tako, da sta prvi števki rojstni dan v mesecu, drugi števki predstavljata mesec, naslednje tri pa leto brez tisočice. Prvih sedem števk EMŠO osebe, rojene 10. 5. 1983, bi bil 1005983. Prvih sedem števk EMŠO osebe, rojene 2. 3. 2001, pa bi bil 0203001. Ali je oseba rojena leta 1xxx ali 2xxx, sklepamo po stotici.

Primer

Recimo, da so v datoteki emso.txt zapisane naslednje številke.

```
1302935505313
1002968501123
1302003500231
2110987505130
1302999350538
2110912501130
```

Funkcija mora tedaj delovati takole.

```
>>> histogramDni("emso.txt")
10. 2. 1
13. 2. 3
21.10. 2
```

5. Stirlingova števila drugega reda

Napišite funkcijo `stirling2(n)`, ki izpiše n vrstic trikotnika Stirlingovih števil drugega reda. Trikotnik sestavimo po naslednjem pravilu: prvi in zadnji element vsake vrstice sta 1. Elemente vmes dobimo tako, da seštejemo element levo zgoraj in k-krat element zgoraj, pri čemer je k številka stolpca.

```
1
1    1
1    3    1
1    7    6    1
1   15   25   10    1
1   31   90   65   15    1
1   63  301  350  140   21    1
1  127  966 1701 1050  266   28    1
```

Primer: 90 je v tretjem stolpcu; naračunali smo jo kot 15 + 3×25. 350 je v četrtem stolpcu, dobili smo jo kot 90+4×65.