Izpit iz predmeta Programiranje, 9. februar 2012

Rešitve **vseh** nalog shranite **v eno samo datoteko s končnico .py** in jo oddajte prek Učilnice tako kot domače naloge. Vse funkcije naj imajo enaka imena in argumente, kot jih predpisuje naloga. Rezultate naj vrnejo, ne izpišejo. Pri reševanju nalog je dovoljena vsa literatura na poljubnih medijih, ves material, ki je objavljen na Učilnici, vključno z objavljenimi programi; njihova uporaba in predelava se ne šteje za prepisovanje.

Pozorno preberi naloge in ne rešuj le na podlagi podanih primerov! Da rešitev ne bi imela trivialnih napak, jo preverite s testi v ločeni datoteki na Učilnici. Za rešitev naloge lahko dobite določeno število točk, tudi če ne prestane vseh testov. Funkcija, ki prestane vse teste, še ni nujno pravilna.

Izpit morate pisati na fakultetnih računalnikih, ne lastnih prenosnikih.

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

Čas pisanja: 90 minut.

Nalogi A in B se ne točkujeta, temveč sta **obvezni** za uspešno opravljen izpit. Kdor ju ne reši pravilno, je s tem že padel. Nalogi morate rešiti tako, da predelate SVOJO rešitev domače naloge. Samo tisti, ki naloge niso oddali, oziroma jim je bila ocenjena negativno, naj vzamejo objavljeno rešitev.

Ostale naloge so vredne enako število točk.

A. Osem napadalnih kraljic

Spremeni funkcijo `prost_stolpec`, tako da bo vrnila prvi prosti stolpec iz desne proti levi. Če so, recimo, prosti stolpci a, b in e, naj funkcija vrne stolpec e in ne a, tako kot prej.

B. Funkcije za delo z vektorji

Dodaj funkcijo `iclear(xs)`, ki vse elemente podanega vektorja `xs` nastavi na 0. Funkcija naj vrne `None`.

1. Skritopis

Napiši funkcijo `skritopis(s)`, ki (kar predobro) skrije besedilo tako, da vsako besedo zamenja z njeno prvo črko. Vse ostale znake (presledke, ločila) pusti, kot so. Funkcija naj kot argument sprejme niz in kot rezultat vrne skriti niz.

Če ti ne uspe rešiti naloge, kot je opisana, pa predpostavi, da v stavku ni ločil, temveč le besede, ločene s presledki. V tem primeru dobiš le polovico točk.

Namig: če hočeš rešiti celotno nalogo, NE uporabi metode split, ker ne naredi ničesar koristnega!

Primer

```
>>> skritopis("Napiši funkcijo skritopis(s), ki (kar predobro) skrije besedilo tako,
da vsako besedo zamenja z njeno prvo črko.")
'N f s(s), k (k p) s b t, d v b z z n p č.'
```

2. Odvečni presledki

Nekateri površneži v besedilo dodajajo nepotrebne (odvečne) presledke med besedami. Napišite funkcijo `prepresledki(s)`, ki prejme tak niz in izračuna delež presledkov, ki so brez potrebe "podaljšani". "Podaljšan presledek" je vsako zaporedje presledkov, ki vsebuje dva ali več presledkov; pri tem se četverni presledek šteje kot en podaljšani presledek.

Primer

```
>>> prepresledki("Tule je      samo en preveč.")
0.25
>>> prepresledki("Tule ni nobenega preveč.")
0.0
>>> prepresledki("Ta    je  res      pretiran.")
```

```
1.0
>>> prepresledki("Problem!")
0.0
```

3. Objektni subjekti

Imamo razred `Oseba`, ki je definiran takole:

```
class Oseba:
    def __init__(self, ime, spol, emso):
        self.ime = ime
        self.spol = spol
        self.emso = emso
```

Pri tem je spol lahko M ali Ž.

Napiši metodo `preveri_emso`, ki preveri pravilnost EMŠO in vrne `True`, če je pravilna in `False`, če ni. EMŠO je pravilna, če ima smiseln datum (nihče ni rojen 35. maja ali 31. aprila; predpostaviti smeš, da imajo vsi februarji 29 dni), se spol ujema s podanim spolom in je kontrolna številka izračunana pravilno.

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 med 500 in 999 za ženske. Kontrolna številka je izračunana takole: prvo števko EMŠO pomnožimo s 7, drugo s šest, tretjo s pet in tako naprej, do šeste, ki jo pomnožimo z 2. Sedmo spet pomnožimo s 7, osmo s 6, deveto s 5 in tako do dvanajste, ki jo pomnožimo z 2. Za zadnjo, trinajsto številko, velja tole: če jo prištejemo h gornji vsoti, dobimo število, ki je deljivo z 11.

4. Poet tvoj nov Slovencem venec vije

Predpostavimo, da se dve besedi rimata, če se ujemata v zadnjem zlogu. Napiši funkcijo, ki prejme kitico pesmi v obliki večvrstičnega niza: vsaka vrstica pesmi je ena vrstica niza. Besede so ločene s presledki, v besedilu so tudi vejice in druga ločila. Funkcija naj vrne niz, ki pove, kakšne oblike je rima. To določimo tako, da

- vrstici, ki se ne rima z nobeno predhodno vrstico, priredimo novo črko; črke jemljemo po abecedi
- vrstici, ki se rima s predhodno, priredimo enako črko kot tej vrstici.

Spodaj je nekaj primerov.

Poét tvoj nòv Slo-ven-cem vê-nec ví-je, 'z pet-nájst so-ne-tov ti ta-kó ga splê-ta, de »Mà-gí-strá-le«, pe-sem tri-krat pé-ta, vseh dru-gíh sku-paj vé-že hà-rmo-ní-je.	Tí si žív-ljê-nja moj-'ga mà-gí-strá-le, gla-síl se 'z njê-ga, kò ne bó več mê-ne, ran mô-jíh bò spo-mín ín tvô-je hva-le.	To ni pe-sem, so le ne-ke pi-sa-ri-je, kar brez ri-me, (če se kam ne skri-je).
Rima: ABBA	Rima: ABA	Rima: ABCA

5. Turnir

Turnirska pravila starodavne angleške igre Evenzero so naslednja:

- na turnirju se vedno pomeri 2^n (torej 1, 2, 4, 8, 16...) tekmovalcev,
- v vsakem krogu se pomeri prvi tekmovalec z drugim, tretji s četrtim, peti s šestim in tako naprej. Zmagovalci izmed teh parov se uvrstijo v naslednji krog.

Igra je preprosta: tekmovalca preštejeta, kolikšna je vsota dolžin njunih imen. Če je soda, zmaga prvi tekmovalec, če liha, drugi. Napišite funkcijo, ki kot argument prejme seznam tekmovalcev in kot rezultat vrne ime zmagovalca.

Primer

```
>>> turnir(['Alice', 'Bob', 'Tom', 'Judy'])
'Judy'
```

V prvem krogu tekmujeta Alice in Bob (zmaga prvi, torej Alice, ker je vsota dolžin njunih imen, osem, soda) in Tom in Judy (zmaga drugi, Judy, ker je dolžina imen liha). V drugem krogu tekmujeta Alice in Judy; zmaga drugi, Judy, saj je dolžina liha.