

Programiranje 1, praktična matematika

Poskusni izpit 2

1. [3 × 2 točki] Sestavi izraze (ne stavke), ki imajo za vrednosti [predpostavi, da je ustrezen modul že uvožen]:

- Vsoto vseh vrednosti (ki so zagotovo števila) v slovarju sl

- vsoto dveh naključnih sodih 2-mestnih naravnih števil

- večje od dveh decimalnih števil, zapisanih v prvih dveh vrsticah ravnokar za branje odprte datoteke dat.

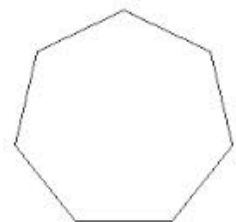
2. [3 × 5 točk] Sestavi preproste funkcije:

- Funkcija, ki nam vrne število vrstic, v katerih imamo sodo število znakov.

- Funkcija, ki »obrne« slovar, katerega vrednosti so bodisi posamezne vrednosti, bodisi sezname posameznih vrednosti:

{1: [1,2,3,4,5],2: 4,3: [4,1,2,"a"]} -> {1: [1, 3], 2: [1, 3], 3: [1], 4:[1, 2, 3], 5:[1], 'a': [3]}

- Funkcija, ki nariše n -kotnik s stranicamo dolžine a . Pri tem mora biti $a > 3$, drugače vrzi izjemo.



3. Dana je definicija funkcije f z enim parametrom.

```
def f(x):  
    if x < 2: return x  
    return x % 2 + f(x // 2)
```

Kakšne so vrednosti naslednjih izrazov:

$f(0)$ _____

$f(1)$ _____

$f(5)$ _____

$f(11)$ _____

$f(16)$ _____

Kaj funkcija f vrne kot rezultat?

a) potenco števila 2 v razcepu števila x na prafaktorje

b) število enic v dvojiškem zapisu števila x

c) število deliteljev števila x

d) drugo: _____

4. [8 točk] Sestavi funkcijo, ki v izbrani mapi (in njenih podmapah) poišče mapo, ki vsebuje največ datotek. Funkcija naj vrne ime iskane mape (skupaj s potjo) ter število datotek v tej mapi

5. [5 točk] V imeniku imamo mape, ki vsebujejo .txt datoteke, v katerih so nerešene naloge iz matematike. Sestavi funkcijo, ki bo rešila vse naloge in ustrezne računa zapisala na izhodno datoteko. Vsak račun je v svoji vrstici. Datoteka je sestavljena tako:

```
4 + 5
2*5 - 10
...
```

Ko funkcija naloge reši, mora izhodna datoteke zgledati takole:

```
4 + 5 = 9
2*5 - 10 = 0
...
```

6. [8 točk] Sestavi metodo `brezOklepajev(imeVhod, imeIzhod)`

ki tekstovno datoteko, katere ime je v `imeVhod` prepíše na novo datoteko z imenom, kot ga določa `imeIzhod`. Pri tem naj iz vrstic, ki vsebujejo izraze v oklepajih, pobriše te izraze skupaj z oklepaji. V vsaki vrstici se lahko pojavi največ en par pravilno postavljenih oklepajev. V primeru, da vrstica ni ustrezno sestavljena, jo v izhodni datoteki nadomesti z vrstico "NAPAKA" Primer:

<i>Vhodna datoteka</i>	<i>Izhodna datoteka</i>
Peter (spet) zamuja.	Peter zamuja.
Jaz sem (napisana narobe	NAPAKA
To ni račun.	To ni račun.
2*2+(2-7)=8	2*2+=8
To) (ni prav.	NAPAKA
[(2)]	[]
Konec(.).	Konec.

Nasvet: napiši primerne pomožne funkcije.

6. **[10 točk]** Sestavi funkcijo `frekvencnaAnalizaProstor(datoteka)`, ki za dano datoteko prešteje, kolikokrat se v njej pojavi katera črka. Omejiš se lahko na črke angleške abecede. Funkcija naj ne ločuje med malimi in velikimi črkami. Vse znake, ki niso črke, naj ignorira. Funkcija naj vrne seznam števcov, ki ustrezajo frekvencam posameznih črk. Vemo, da celotna vsebina datoteke ne gre naenkrat v pomnilnik

7. **[10 točk]** Levenshteinova razdalja oz. urejevalniška razdalja med nizoma a in b je najmanjše število elementarnih operacij, ki so potrebne, da niz a spremenimo v niz b . Elementarne operacije so:

- brisanje enega znaka iz niza;
- vstavitve enega novega znaka v niz;
- zamenjava obstoječega znaka z novim.

Sestavite funkcijo `levenshteinovaRazdalja(a, b)`, ki kot argumenta prejme dva niza: `a` in `b`. Funkcija naj vrne urejevalniško razdaljo med `a` in `b`.

```
>>> levenshteinovaRazdalja('banana', 'ananas')
2
>>> levenshteinovaRazdalja('potop', 'pokol')
2
>>> levenshteinovaRazdalja('matematika', 'fizika')
7
```