

Izpit

Tekom izpita boste napisali svoj modul. Poimenujte ga **Ime_Priimek**. Ko končate z izpitom ga pošljite na gregor.balkovec@ltfe.org in anze.glusic@ltfe.org.

Naloga 01

Točka: / 5

Napišite funkcijo `funkcija01`, ki kot prvi parameter sprejme višino valja in kot drugi parameter sprejme radij valja. Funkcija naj vrne izračunano prostornino valja (lahko privzamete, da so vse volumenske enote v centimetrih in da je izhod v kubičnih centimetrih).

" π " najdete v built-in knjižnici **math**.

```
import math
print(math.pi)
INPUT:
funkcija01(2, 3)
```

OUTPUT:
56.54866776

In []:

In []:

Naloga 02

Točke: / 5

Napišite funkcijo `funkcija02`, ki prejme list.

Vsak element v listu je tuple. Prvi element v tuplu je ocena, katero je študente prejel na izpitu. Drugi element je ime študenta.

Funkcija naj vrne list imen študentov, ki so imeli oceno 50 ali več.

```
INPUT:
funkcija02([[50, "Jan"], [49, "Anže"], [74, "Tine"], [100, "Anja"], [12, "Zvone"], [81, "Mirko"]])
OUTPUT:
['Jan', 'Tine', 'Anja', 'Mirko']
```

In []:

In []:

Naloga 03

Točke: /5

Napiši funkcijo `funkcija03`, ki prejme 1 parameter - dictionary, ki predstavlja cene kriptovalut v evrih na dva različna dni.

Dictionary je sestavljen na sledeč način:

- Prvo je razdeljen na 2 ključa, ki predstavljata dva dni.
- Vsak dan ima kot vrednost nov dictionary
- - Znotraj imamo imena različnih kriptovalut, njihovih cen v evrih na določen dan, in njihov market cap v evrihi na določen dan

Funkcija naj sedaj izračuna za koliko procentov se je določena kriptovaluta spremenila iz prvega dneva v drugi dan in to izpiše v obliki:

KOVANEC se je spremenil za SPREMEMBA %

Formula za izračun relativnega povečanja je sledeča:

$$\frac{cena_{dan2} - cena_{dan1}}{cena_{dan1}}$$

Funkcija naj nato vrne ime kovanca, ki se je povečal za največ procentov.

INPUT:

`funkcija03(data)`

OUTPUT:

bitcoin se je spremenil za 0.04 %

pivx se je spremenil za -0.22 %

polkadot se je spremenil za -0.10 %

ethereum se je spremenil za 0.37 %

cardano se je spremenil za 0.17 %

'ethereum'

```
In [1]: data = {
    "day_1": {
        "pivx": {
            "eur": 0.466608,
            "eur_market_cap": 31703850.28872451
        },
        "bitcoin": {
            "eur": 41653,
            "eur_market_cap": 789077487998.7858
        },
        "cardano": {
            "eur": 1.08,
            "eur_market_cap": 34819120071.59348
        },
        "polkadot": {
            "eur": 22.09,
            "eur_market_cap": 23660724367.996834
        },
        "ethereum": {
            "eur": 3382.92,
            "eur_market_cap": 403045423232.84467
        }
    },
    "day_2": {
```

```

    "bitcoin": {
      "eur": 43153,
      "eur_market_cap": 789077487998.7858
    },
    "pivx": {
      "eur": 0.365668,
      "eur_market_cap": 31703850.28872451
    },
    "polkadot": {
      "eur": 19.85,
      "eur_market_cap": 23660724367.996834
    },
    "ethereum": {
      "eur": 4624.21,
      "eur_market_cap": 403045423232.84467
    },
    "cardano": {
      "eur": 1.26,
      "eur_market_cap": 34819120071.59348
    },
  },
}

```

In []:

In []:

Naloga 04

Točke: /5

Napišite funkcijo `funkcija04`. Funkcija naj odpre datoteko `input_file.txt`. Pregleda naj vsako vrstico in preveri koliko črk se nahaja v vrstici. Nato naj vrstice razporedi od najkrajše do najdaljše in jih v takšenm vrstnem redu zapiše v datoteko `output_file.txt`.

INPUT:

```

His palms are sweaty, knees weak, arms are heavy
There's vomit on his sweater already, mom's spaghetti
He's nervous, but on the surface he looks calm and ready
To drop bombs, but he keeps on forgettin'
What he wrote down, the whole crowd goes so loud
He opens his mouth, but the words won't come out
He's chokin', how, everybody's jokin' now
The clocks run out, times up, over, blaow
Snap back to reality, ope there goes gravity
Ope, there goes Rabbit, he choked
He's so mad, but he won't give up that easy? No
He won't have it, he knows his whole back's to these ropes
It don't matter, he's dope, he knows that, but he's broke
He's so stagnant, he knows, when he goes back to this mobile home,
that's when it's
Back to the lab again, yo, this whole rhapsody
Better go capture this moment and hope it don't pass him

```

OUTPUT:

```

Ope, there goes Rabbit, he choked
He's chokin', how, everybody's jokin' now
The clocks run out, times up, over, blaow
To drop bombs, but he keeps on forgettin'

```

```
Snap back to reality, ope there goes gravity
Back to the lab again, yo, this whole rhapsody
He's so mad, but he won't give up that easy? No
He opens his mouth, but the words won't come out
His palms are sweaty, knees weak, arms are heavy
What he wrote down, the whole crowd goes so loud
There's vomit on his sweater already, mom's spaghetti
Better go capture this moment and hope it don't pass him
He's nervous, but on the surface he looks calm and ready
It don't matter, he's dope, he knows that, but he's broke
He won't have it, he knows his whole back's to these ropes
He's so stagnant, he knows, when he goes back to this mobile home,
that's when it's
```

In []:

In []:

Naloga 05

Točke: /5

Napišite razred `Planet`. Ko ustvarimo novo instanco razreda, vanj shranimo ime, razdaljo od sonca in maso planeta.

`Planet` naj ima tudi funkcijo `razdalja_od_planeta`, ki kot argument prejme še en drugi planet. Funkcija naj izračuna kolikšna je razdalja med planetoma.

Napišite tudi razred `Osoncje`. Ko ustvarimo novo instanco razreda lahko kot argument pošljemo list planetov ali pa tudi ne.

`Osoncje` naj ima spremenljivko `planeti`, ki je list planetov, ki se nahajajo v osončju.

`Osoncje` naj ima funkcijo `dodaj_planet`, ki kot argument prejme planet in le-tega doda v svoj list planetov.

`Osoncje` naj ima funkcijo `planet_z_najvecjo_maso`, ki naj vrne ime planeta z največjo maso.

`Osoncje` naj ima funkcijo `razvrsti_po_oddaljenosti`, ki naj planete razvrsti od najblizjega do najblj oddaljenega planeta. Nato naj funkcija vrne list samo imen planetov v pravilnem vrstnem redu.

```
venera = Planet("Venera", 108_000_000, 40.8685)
zemlja = Planet("Zemlja", 152_000_000, 59.742)
uran = Planet("Uran", 19_000_000_000, 868)
```

```
print(zemlja.razdalja_od_planeta(venera))
OUTPUT:
44000000
```

```
print(zemlja.razdalja_od_planeta(uran))
OUTPUT:
18848000000
```

```
mars = Planet("Mars", 227_000_000, 6.4185)
jupiter = Planet("Jupiter", 816_000_000, 18980)
```

```
osoncje = Osoncje(planeti=[venera, zemlja, uran, mars, jupiter])
```

```
print(osoncje.razvrsti_po_oddaljenosti())
```

OUTPUT:

```
['Venera', 'Zemlja', 'Mars', 'Jupiter', 'Uran']
```

```
saturn = Planet("Saturn", 1_513_000_000, 5684)
```

```
merkur = Planet("Merkur", 68_000_000, 3.3022)
```

```
neptun = Planet("Neptun", 30_000_000_000, 102)
```

```
osoncje.dodaj_planet(saturn)
```

```
osoncje.dodaj_planet(merkur)
```

```
osoncje.dodaj_planet(neptun)
```

```
print(osoncje.razvrsti_po_oddaljenosti()) ==> ['Merkur', 'Venera',  
'Zemlja', 'Mars', 'Jupiter', 'Saturn', 'Uran', 'Neptun']
```

```
print(osoncje.planet_z_najvecjo_maso()) ==> Jupiter
```

In []:

In []: