Programmeerimise alused II

Eksami näidisülesanded

Sügis 2016

Eksami korraldus

- Eksam kestab 120 minutit (paberosa + arvutiosa).
- Paberosa
    - ilma arvutita
    - ilma materjalideta
    - ilma suhtlemiseta
    - töö äraandmisaja otsustab üliõpilane
    - max 20 punkti, positiivse hinde jaoks min 10 punkti

    Kõiki vastuseid tuleb põhjendada! Veateate korral pole vajalik selle täpse teksti esitamine, vaid põhjuse kirjeldamine.

    Kõrgelt hinnatakse õigeid vastuseid ja selgitusi. Samas omavad väärtust ka selgitused, mis näitavad mõistlikku mitmevahelolekut.

- Arvutiosa
    - arvutiga
    - materjale võib kasutada
    - ilma suhtlemiseta
    - töö esitamine Moodle'is
    - max 20 punkti, positiivse hinde jaoks min 10 punkti

Paberosa näiteid

Ülesanne (? punkti)

Järgnev programmilõik leiab kahemõõtmelise järjendi korral, kui paljudes ridades on positiivseid elemente.

```
positiivsegaRidu = 0
for i in range(len(a)):
     leidubPositiivne = False
     for j in range(len(a[i])):
          if a[i][j] > 0:
               leidubPositiivne = True
               break
     if leidubPositiivne:
          positiivsegaRidu += 1

print(positiivsegaRidu)
```

Koostada funktsioon `onPositiivseid`, mille puhul alltoodud programmilõik töötaks ülaltooduga võrdväärselt.

```
positiivsegaRidu = 0
for i in range(len(a)):
     if onPositiivseid(a[i]):
          positiivsegaRidu += 1
print(positiivsegaRidu)
```

Ülesanne (? punkti)

Mida väljastatakse programmi töötamisel ekraanile?

```
arva = 4
arvb = arva
arvb = 7
print(arva)
lista = [-1, 4, 5, -2]
listb = lista
listb[1] = 7
print(lista[1])
listc = lista[:]
lista[2] = 8
print(listc[2])
```

Ülesanne (? punkti)

Mida väljastatakse programmi töötamisel ekraanile?

```
muusikud = ("John", "Paul", "George", "Ringo")
a = [1941, 1842, 1943]
a[1] = 1942
s = {muusikud[0]: a[0], muusikud[2]: a[2], muusikud[3]: a[0]}
s["Paul"] = a[1]
print(s["John"])
print(s["Paul"])
s2 = s.copy()
del s2["John"]
s.update({"John": 1940})
for i, j in s2.items():
    print(j, i)
```

Ülesanne (? punkti)

Mida väljastatakse programmi töötamisel ekraanile?

```
h1 = {1, 2 ,3 ,4}
h2 = {3, 5, 6, 7}
h2.add(7)
h3 = {4, 6, 7}
h4 = h1 | h2 & h3
print(h4 ^ {1, 2, 5})
```

Ülesanne (? punkti)

Mida trükib ekraanile järgmine programmilõik?

```
def rekFun(x, y):
    print(x)
    if y < 0:
        return x
    else:
        return rekFun(x + 2, y - 1) + rekFun(x + 4, y - 1)
print(rekFun(-2, 1))
```

Arvutiosa näiteid

Ülesanne (? punkti)

Kirjutada funktsioon, mis võtab argumendiks 2 maatriksit ning kontrollib, kas üks on saadud teisest transponeerimise teel. (Maatriksid ei pruugi olla ruutmaatriksid.)
https://et.wikipedia.org/wiki/Transponeeritud_maatriks

Näide tööst:

Maatriksite
```
maatriks_1 = [[1, 2], [3, 4], [5, 6]]
maatriks_2 = [[1, 3, 5], [2, 4, 6]]
```
puhul peaks funktsioon tagastama **True**

Kutsudes välja funktsiooni maatriksitega
```
maatriks_3 = [[1, 2], [1, 1]]
maatriks_4 = [[1, 2, 3], [2, 2, 3]]
```
peaks funktsioon tagastama **False.**

Ülesanne (? punkti)

Kirjutada funktsioon *tulba_pikimad*, mis võtab argumendiks kahemõõtmelise järjendi, mille elementideks on sõnad. Funktsioon leiab igas tulbas pikima sõna ning tagastab listi nendest sõnadest. Sama pikkusega sõnade puhul võetakse eespool olev sõna.

Kirjutada funktsioon *enim_taishaalikuid*, mis võtab argumendiks sõnade listi ning tagastab sõna, milles on kõige rohkem täishäälikuid. Täishäälikute leidmiseks võib kasutada listi:
`taishaalikud = ["a","e","i","o","u","õ","ä","ö","ü"]`

Kasutades neid funktsioone, leida kahemõõtmelise järjendi tulba pikimatest sõnadest sõna, milles on kõige rohkem täishäälikuid.

Näide tööst:

```
sonade_tabel = [["piim", "kommikarp", "lillkapsas"], ["leib",
"hakkliha", "sink"], ["makaronid", "mandariinid", "tee"]]
```

Sellise järjendi puhul peab funktsioon tagastama:
`mandariinid`

Ülesanne (? punkti)

Pekingi olümpiamängude tarbeks rakendati 2008. aastal ajutiselt liikluses regulatsioonid, et vähendada õhusaastet. Paarisarvulisel kuupäeval võisid liikuda vaid autod, mille numbrimärgil oli paarisarv ning paaritul kuupäeval võisid liikuda autod, mille numbrimärgil oli paaritu arv.

Failis on igal real ühe pere autode numbrimärgid tühikuga eraldatud.
Küsida kasutajalt kuupäev, kujul DD-MM-YYYY.
Funktsioon võtab argumentideks failist loetud kahemõõtmelise järjendi ja kasutaja sisestatud kuupäeva ning tagastab perede arvu, kes ei saa sel päeval autoga liikuda.

Näide tööst:
Näidisfail: https://drive.google.com/open?id=0BzdoAawekCh-NDZzejBjUlNwVDA

Sellise näidisfailiga peaks kuupäev `14-detsember-2016` andma vastuseks `3`, ning kuupäev `15-detsember-2016` andma vastuseks `4`.

Ülesanne (? punkti)

Kirjutada funktsioon, mis võtab argumendiks sõnade järjendi ning tagastab sõnastiku, kus võtmeks on sõna ning väärtuseks sõnas esinevate täishäälikute arv.

Näide tööst:

```
sonade_jarjend = ["logistik", "bussijuht", "autolukksepp",
"kirjakandja", "programmeerija", "klienditeenindaja"]
```

Sellise järjendi puhul peaksid sõnastikus olema allolevad elemendid:

```
{'kirjakandja': 4, 'logistik': 3, 'programmeerija': 6,
'klienditeenindaja': 8, 'autolukksepp': 5, 'bussijuht': 3}
```

Rahvusvaheliste sõidukite ülesanne

Ülesanne (? punkti)

Python lubab listidesse panna liste, mille elementideks on omakorda listid, mille elementideks on listid jne.

Kirjuta funktsioon max_pikkus, mis võtab argumendiks listi, mille elementideks võivad olla täisarvud või listid, mille elementideks võivad olla täisarvud või listid, mille jne. Funktsioon peab tagastama pikima selles puntras leiduva listi pikkuse.

Näited:

- `max_pikkus([1,2,3])` peab tagastama *3*
- `max_pikkus([[[1,2,3]]])` peab samuti tagastama *3* (kõige sisemise listi pikkus on 3)
- `max_pikkus([[], [3,[4,5],[2,3,4,5,3,3], [7], 5, [1,2,3], [3,4]], [1,2,3,4,5]])` peab tagastama 7
  (listil `[3,[4,5],[2,3,4,5,3,3], [7], 5, [1,2,3], [3,4]]` on 7 elementi)

Ülesanne (? punkti)

Koostada rekursiivne funktsioon, mis saab argumendina ühe positiivse täisarvu. Funktsioon peab (igal väljakutsel) ekraanile väljastama argumendi. Rekursiivselt peab korraldama, et igal järgmisel väljakutsel on argument eelmisest 3 võrra väiksem ja argument oleks positiivne. Lõpuks peab funktsioon peab tagastama esimese sellise arvu, mis pole positiivne.

Kasutamise näited

```
print("Tagastatakse: " + str(fun(11)))
```

11
8
5
2
Tagastatakse: -1

```
print("Tagastatakse: " + str(fun(6)))
```

6
3
Tagastatakse: 0

```
print("Tagastatakse: " + str(fun(2)))
```

2
Tagastatakse: -1

```
print("Tagastatakse: " + str(fun(-3)))
```

Tagastatakse: -3

Paarissumma ülesanne