

# Programmeerimise 1. vaheksam

Neljapäev, 30.10.2014, kell 10:15 – 12:00

## 1. Sulgude paarsuse kontroll (5p) yl1.py

Kirjuta funktsioon `on_paaris`, mis võtab argumentiks sõne, ja tagastab `True` või `False` vastavalt sellele, kas antud sõne koosneb omavahel paaris olevatest ümarsulgudest või mitte. Näide:

```
>>> on_paaris("(")
False
>>> on_paaris("()")
False
>>> on_paaris("()()")
False
>>> on_paaris(")(")
False
>>> on_paaris("()")
True
>>> on_paaris("(()())")
True
```

Vihje: sõnet läbi käies pea arvet, mitu avatud sulgu on antud kohas sulgemata.

## 2. Valimistulemused (10p) yl2.py

Failis `valimistulemused.txt` on valimiste tulemused, kus igal real on ühe kandidaadi nimi, koolon, talle antud häälte arv, kaldkriips ning tema valimisringkonnas hääletanute arv. Faili read on järjestatud häälte arvu järgi kahanevalt, st. esimesel real on kõige enam hääli saanud kandidaat.

Kirjuta programm, mis väljastab kahes erinevas järjestuses ekraanile nende kandidaatide nimed, kes said rohkem kui 10% oma valimisringkonna häältest. Esmalt tuleb kuvada nimed perenime järgi tähestiku järjekorras, siis kuvada tühi rida ja seejärel samad nimed kogutud häälte järjekorras (st samas järjekorras nagu need failis paiknesid).

Näide. Kui `valimistulemused.txt` sisu on

```
Kikas, Tiit: 740/5040
Prits, Priit: 603/5040
Maasikas, Malle: 250/3112
Kartul, Kalle: 145/5639
Paun, Peeter: 139/500
```

siis programm peaks väljastama

```
Kikas, Tiit
Paun, Peeter
Prits, Priit

Kikas, Tiit
Prits, Priit
Paun, Peeter
```

Vihje:

```
>>> sorted(["cda", "aab", "abc"])
['aab', 'abc', 'cda']
```

Programmis peab olema funktsioon `eralda`, mis võtab argumentiks sõne, mis kujutab ühe isiku valmistulemuse rida, ning tagastab listi, kus esimene element on inimese nimi, teine element tema poolt antud häälte arv (täisarvuna) ja kolmas tema häälte arvu ja valimisringkonnas hääletanute suhe (ujukomaarvuna). Näide:

```
>>> eralda("Kartul, Kalle: 145/5693\n")
["Kartul, Kalle", 145, 0.025469875285438256]
```

---

Programmid peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). **Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma.** Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.

### 3. Pulmad ja matused (3+2p) yl3.py

Failis *syndmused.txt* on kirjeldatud kahte sorti sündmuse – pulmad ja matused.

Iga pulma jaoks on failis 3 rida: esimesel real on sõna "Pulm" ning sellele järgneb kuupäev; teisel real on peigmehe nime ning kolmandal real pruudi nimi. Iga matuse jaoks on failis 2 rida: esimesel real on sõna "Matus", millele järgneb kuupäev; teisel real on kadunukese nimi.

#### Ülesande põhiosa (3p)

Kirjuta programm, mis kuvab ekraanile pulmade ja matuste andmed nii, et iga sündmuse info on kokku võetud ühele reale.

Näide. Kui faili *syndmused.txt* sisu on

```
Pulm 12.10.2014
Valdur Vaarikas
Mari Maasikas
Matus 13.10.2014
Jaan Tamm
Matus 14.10.2014
Tiit Torbik
Matus 14.10.2014
Robert Ronija
Pulm 16.10.2014
Siim Suslik
Triin Trikster
```

siis programmi käivitamisel peab ekraanile ilmuma

```
Pulm 12.10.2014, abiellusid Valdur Vaarikas ja Mari Maasikas
Matus 13.10.2014, suri Jaan Tamm
Matus 14.10.2014, suri Tiit Torbik
Matus 14.10.2014, suri Robert Ronija
Pulm 16.10.2014, abiellusid Siim Suslik ja Triin Trikster
```

#### Ülesande lisa (2p)

Programm peaks suutma anda sama väljundi ka siis, kui iga sündmuse kohta käival esimesel real on vaid kuupäev, ilma märketa sündmuse liigi kohta. Sel juhul peab programm sündmuse liigi tuvastama järgnevaid ridu uurides. Seega programm peaks toimima ka selliste failidega:

```
12.10.2014
Valdur Vaarikas
Mari Maasikas
13.10.2014
Jaan Tamm
14.10.2014
Tiit Torbik
14.10.2014
Robert Ronija
16.10.2014
Siim Suslik
Triin Trikster
```

---

*Programmide peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma. Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.*

# Programmeerimise 1. vaheksam

Neljapäev, 30.10.2014, kell 16:15 – 18:00

## 1. Mitu arvu? (5p) yl1.py

Kirjuta funktsioon `naturaalarvude_arv`, mis võtab argumendiks sõne ja tagastab täisarvu, mis näitab, mitu märgita täisarvu leidub antud sõnes. Üheks arvuks loeme alamsõnet, mis koosneb ainult kümnendsüsteemi numbritest ja millele eelneb mõni muu sümbol või sõne algus ja millele järgneb mõni muu sümbol või sõne lõpp. Teisiti öeldes, kõik sümbolid, mis pole numbrid, on arvude eraldajad.

Näited:

```
>>> naturaalarvude_arv("tere")
0
>>> naturaalarvude_arv("323")
1
>>> naturaalarvude_arv("a323b4")
2
>>> naturaalarvude_arv(" 1a323-45 56.7")
5
```

## 2. Keskmised hinded (10p) yl2.py

Failis `hinded.txt` on klassi laste nimekiri koos neile pandud hinnetega. Igal real on kõigepealt õpilase eesnimi, semikoolon, perekonnanimi, semikoolon ja siis komadega eraldatuna tema hinded või puudumised (iga hinne on täisarv 1..5, puudumine on märgitud miinusega).

Kirjuta programm, mis kuvab ekraanile

1. õpilaste nimed, kelle keskmine hinne on väiksem kui 3.0 või kellel on rohkem kui 2 puudumist.
2. keskmise puudumiste arvu õpilase kohta.

Näide. Kui `hinded.txt` sisu on

```
Kaur;Kaupmees;4,5,4,5,-,-,3,2,3,3
Tiit;Tuututaja;-,-,-,3,2,3,2,2,1
Mari;Maasikas;4,4,5,4,3,4,5
Lauri;Laisk;4,4,-,-,5,4,5,5,5,-
Madis;Mardisalu;3,4,2,2,3
```

siis programmi käivitamisel peab ekraanile ilmuma

```
Tiit Tuututaja
Lauri Laisk
Madis Mardisalu
Keskmiselt oli puudumisi 1.8
```

Programmis peab olema defineeritud ja sobivalt kasutatud funktsioon `keskmine_hinne`, mis võtab argumendiks sõne hinnete ja/või puudumismärkidega (kui neid on), ning tagastab kaheelemendilise listi, kus esimene element on keskmine hinne ja teine on puudumiste arv. Võib eeldada, et argumendiks antavas sõnes on vähemalt üks hinne.

Näide:

```
>>> keskmine_hinne("4,5,4,5,-,-,3,2,3,3")
[3.625, 2]
```

---

Programmide peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). **Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma.** Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.

### 3. Maraton (3+2p) *yl3.py*

Failis *tulemused.txt* on jooksuvõistluste andmed. Igal real on ühe võistleja ühe võistluse tulemused kujul:

*Võistleja nimi; võistluse nimi; võistleja aeg minutites*

Read on järjestatud võistleja nime järgi, st ühe võistleja tulemus on järjestikustel ridadel.

#### Ülesande põhiosa (3p)

Kirjuta programm, mis väljastab iga võistleja kohta tema nime ja tema võistluste arvu.

Näide. Kui *tulemused.txt* sisu on

```
Malle Maasikas; Kalevipoja jooksumaraton; 236
Peeter Paun; Elva kevadjooks; 301
Peeter Paun; Kalevipoja jooksumaraton; 326
Priit Prits; Elva kevadjooks; 298
Tiit Kikas; Kalevipoja jooksumaraton; 287
```

siis programmi käivitamisel peab ekraanile ilmuma

```
Malle Maasikas: 1
Peeter Paun: 2
Priit Prits: 1
Tiit Kikas: 1
```

#### Ülesande lisa (2p)

Peale võistlejate info kuvamist tuleb kuvada tühi rida ja seejärel iga failis mainitud võistluse nimi koos parima aja saavutaja nimega. Ülaltoodud näite korral peab programmi kogu väljund olema

```
Malle Maasikas: 1
Peeter Paun: 2
Priit Prits: 1
Tiit Kikas: 1

Kalevipoja jooksumaraton: Malle Maasikas
Elva kevadjooks: Priit Prits
```

---

*Programmid peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma. Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.*

# Programmeerimise 1. vaheksam

Reede, 31.10.2014, kell 8:15 – 10:00

## 1. Pikima sõna pikkus (5p) *yl1.py*

Kirjuta funktsioon `pikima_sõna_pikkus`, mis võtab argumendiks sõne, mis kujutab lauset, ning tagastab selle lause pikima sõna pikkuse. Kirjavahemärkidest võivad esineda punkt, koma ja hüüumärk. Võib eeldada, et kirjavahemärgid esinevad ainult vahetult peale sõna lõppu ja et sõnade vahel või sõnale järgneva kirjavahemärgi ning järgmise sõna vahel on täpselt üks tühik. Sõnale järgnevaid kirjavahemärke me ei loe sõna sümbolite hulka kuuluvaks.

Näited:

```
>>> pikima_sõna_pikkus("Tere tulemast!")
8
>>> pikima_sõna_pikkus("Poesell, Jim, ei öelnud midagi.")
7
```

## 2. Kehaline kasvatus (10p) *yl2.py*

Failis `opilased.txt` on õpilaste nimed koos nende lemmikspordialadega, eelistuse järjekorras. Kui õpilane puudub, siis on tema nime järel märged (`puudub`). Faili formaati vaata täpsemalt allpool olevast näitest.

Kirjuta programm, mis aitab kehalise õpetajal õpilased gruppidesse jagada. Kuna tema enda lemmikala on jalgpall, siis jalgpalli paneb ta mängima kõik kohal olevad õpilased, kelle lemmikute loetelus esineb jalgpall esimesel või teisel kohal. Need õpilased peab programm jagama ka kaheks võimalikult sarnase suurusega võistkonnaks. Ülejäänud kohalolevad õpilased soovib õpetaja koondada ühte gruppi (neil tuleb endale ise tegevus leida).

Näide. Kui `opilased.txt` sisu on

```
Mari Maasikas: hoki, jalgpall, male, kabe
Peeter Paan: jalgpall, male, pesapall
Viktor Viil (puudub): jalgpall, kabe
Tauno Teeperv: jooks, jooga, jalgpall
Priit Poolik: male
Kadri Kaunis: jalgpall, hoki
Sander Sats: suusatamine
```

siis programmi käivitamisel peab ekraanile ilmuma midagi taolist (jalgpallivõistkondade täpne jaotus pole oluline):

```
Jalgpallivõistkond A:
Mari Maasikas

Jalgpallivõistkond B:
Peeter Paan
Kadri Kaunis

Ülejäänud:
Tauno Teeperv
Priit Poolik
Sander Sats
```

Programmis peab olema defineeritud ja sobivalt kasutatud funktsioon `sobib_jalgpalliks`, mis võtab argumendiks eelistuste loetelu (sõnede listina) ning tagastab `True` või `False` vastavalt sellele, kas õpetaja soovib selliste eelistustega õpilast jalgpalli mängima või mitte.

Näide:

```
>>> sobib_jalgpalliks(["hoki", "jalgpall", "male", "kabe"])
True
```

*Vihje: Failist andmete lugemisel eralda üksteisest kõigepealt nimi ja hobide loetelu.*

---

*Programmide peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhuga jaoks. Kui funktsiooniga ei oska, siis tee ilma. Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.*

### 3. Erinevad read (3+2p) yl3.py

#### Ülesande põhiosa (3p)

Antud on kaks faili: *a.txt* ja *b.txt*.

Kirjuta programm, mis võrdleb neid faile. Kui failide sisu on erinev, siis tuleb kuvada reanumbrid, kus failide sisu on erinev. Kui ühes failis on vähem ridu, siis tuleb muuhulgas kuvada kõik need reanumbrid, mis ühes failis eksisteerivad ja teises mitte. Kui failid on identsed, siis ei tule kuvada midagi.

Näide. Kui failide sisud on

<i>a.txt</i>	<i>b.txt</i>
kapsas peet	kapsas porgand peet

siis programmi käivitamisel peab ekraanile ilmuma

```
2
3
```

#### Ülesande lisa (2p)

Kui failide sisud on erinevad, aga need saaks teha võrdseks ühest failist ühe rea eemaldamisega, siis tuleb eespool nõutule lisaks kuvada tühi rida ja seejärel vastava faili nimi ja rea number. Vastasel juhul tuleb kuvada tühi rida ja seejärel tekst **Failide ühtlustamiseks rea eemaldamisest ei piisa**.

Eespool toodud näitefailide korral peaks programmi kogu väljund olema

```
2
3
b.txt
2
```

# Programmeerimise 1. vaheksam

Reede, 31.10.2014, kell 10:15 – 12:00

## 1. Mitu topeltvokaali? (5p) yl1.py

Kirjuta funktsioon `topeltvokaalide_arv`, mis võtab argumendiks sõne ja tagastab täisarvu, mis näitab mitu topeltvokaali on antud sõnes. Kolme- või enamakordseid sama vokaali järjendeid ei loe me topeltvokaaliks. Vokaalideks loeme siin ainult eesti keele vokaale (a, e, i, o, u, õ, ä, ö, ü). Lihtsuse mõttes eeldame praegu, et sõnes ei ole suuri tähti.

Näited:

```
>>> topeltvokaalide_arv("kapsas")
0
>>> topeltvokaalide_arv("piiripiiga")
2
>>> topeltvokaalide_arv("puuõõs")
2
>>> topeltvokaalide_arv("bäää!")
0
```

Vihje:

```
>>> s = "kala"
>>> x = 1
>>> s[x]
'a'
>>> s[x-1]
'k'
>>> s[x+1]
'l'
```

*Lihtsustus: Kui sa ei oska eristada topeltvokaali kolmekordsest vokaalist, siis loenda lihtsalt need kohad, kus mitu sama vokaali on kõrvuti.*

## 2. Aknapesijad (10p) yl2.py

Failis `personal.txt` on aknapesuga tegeleva firma töötajate andmed. Igal real on töötaja nimi, tema kehakaal ja tööstaaž (kuud või aastad ja kuud). Read on järjestatud kaalude järgi nii, et esimesel real on kõige kergem töötaja ja viimasel kõige raskem. Kuna firmal on väikese võimsusega tõstukid, siis soovib töödejuhataja moodustada paarid nii, et kõige raskem töötaja on paaris kõige kergemaga jne. Kui sellise jaotamisega jääb nimekirja keskmine inimene ilma paariliseta, siis tema tööle ei pea minema. Lisaprobleem on see, et täna on ilm väga tuuline, seega otsustab juhataja valikusse kaasata vaid need töötajad, kelle tööstaaž on vähemalt 6 kuud.

Näide. Kui `personal.txt` sisu on

```
Peeter Paan; 12kg; 45a,3k
Mari Maasikas; 46kg; 5k
Leo Lill; 67kg; 1a,4k
Jaan Juurikas; 88kg; 2a,3k
Joosep Joodeldaja; 98kg; 4a,8k
Suur Peeter; 131kg; 7k
```

siis programmi käivitamisel peaks ekraanile ilmuma

```
Peeter Paan Suur Peeter
Leo Lill Joosep Joodeldaja
```

Programmis peab olema defineeritud ja sobivalt kasutatud funktsioon `kuude_arv`, mis võtab argumendiks sõne kujul `<aastad>a, <kuud>k` või `<kuud>k` (nt. `3a,4k` või `7k`), ning tagastab täisarvu, mis näitab selle aja pikkust kuudes.

Näited:

```
>>> kuude_arv("3a,4k")
40
>>> kuude_arv("7k")
7
```

---

*Programmid peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhu jaoks. Kui funktsiooniga ei oska, siis tee ilma. Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.*

### 3. Kõik tekstifailid (3+2p) yl3.py

Pythoni `os` moodulis olev funktsioon `listdir` tagastab listi, mis sisaldab näidatud kaustas olevate failide ja alamkaustade nimesid. Jooksva kausta (st. programmi kausta) tähistamiseks võib kasutada punkti.

Näide:

```
>>> import os
>>> os.listdir(".")
['skript_1.py', 'baas.txt', 'katsetus.py', 'lugu.txt', 'andmed.txt']
```

#### Ülesande põhiosa (3p)

Kirjuta programm, mis iga jooksvas kaustas oleva tekstifaili kohta kuvab ekraanile faili nime, faili sisu ja tühja rea. Tekstifailideks loeme faile, mille nime lõpus on `.txt`.

Failid tuleb ekraanile kuvada nende nimede järjekorras.

Võib eeldada, et kõik tekstifailid on kodeeringus UTF-8. Mõned katsetamiseks mõeldud failid leiad Moodle'ist.

```
>>> sorted(['skript_1.py', 'baas.txt', 'katsetus.py', 'lugu.txt', 'andmed.txt'])
['andmed.txt', 'baas.txt', 'katsetus.py', 'lugu.txt', 'skript_1.py']
```

#### Ülesande lisa (2p)

Kui tekstifailides leidub mõni ühine rida (st rida, mis esineb kõigis failides), siis tuleb lõpuks see rida veel eraldi ekraanile kuvada. Vastasel juhul tuleb kuvada `Failides ei leidu ühist rida`.

Näide. Kui jooksvas kaustas on järgmised failid:

<i>skript_1.py</i>	<i>baas.txt</i>	<i>lugu.txt</i>	<i>andmed.txt</i>
<code>print("Tere maailm!")</code>	Tallinn Tartu Tere maailm!	kapsas porgand Tartu	Tartu hernes

siis programmi käivitamisel peaks ekraanile ilmuma

```
andmed.txt
Tartu
hernes

baas.txt
Tallinn
Tartu
Tere maailm!

lugu.txt
kapsas
porgand
Tartu

Tartu
```

---

Programmide peaks küsima kasutajalt vaid neid andmeid, ja väljastama vaid seda, mida ülesanne nõuab (kui üldse on nõutud). **Kui sa mõnda ülesande tingimust ei oska täita, siis lihtsusta ülesannet – näiteks lahenda ülesanne mingi erijuhuga jaoks. Kui funktsiooniga ei oska, siis tee ilma.** Lahendused tuleb salvestada ülesande juures näidatud failinimega ja laadida Moodle'isse. Kindluse mõttes on soovitatav need saata ka enda praktikumijuhendaja e-mailile.