

Programmeerimise 1. vaheeksami järeltöö

Esmaspäev, 23.11.2015, kell 18:15 – 20:00

1. Sõnade arv (7p) *ylesanne1.py*

Kirjuta funktsioon `sõnade_arv`, mis võtab argumendiks sõne, mis koosneb tühikutega eraldatud või üksteise otsa kirjutatud sõnadest ja tagastab sõnade arvu. Selleks, et ka tühikute puudumisel oleks aru saada kus üks sõna lõpeb ja teine algab, lepime kokku, et kõik sõnad algavad suure tähega ja ülejäänud tähed on väikesed.

Näited:

```
>>> sõnade_arv("TöölineJaKolhoositar")
3
>>> sõnade_arv("Tööline JaKolhoositar")
3
>>> sõnade_arv("Tööline Ja Kolhoositar")
3
>>> sõnade_arv("Kolhoositar")
1
>>> sõnade_arv("SiinOnNeliSõna")
4
>>> sõnade_arv("Siin On NeliSõna")
4
```

Vihje:

```
>>> "A".isupper()
True
```

Lihtsustus (-3p): kui suurtähega sõnade eristamine ei õnnestu, siis kirjuta lahendus vähemalt nendele juhtumitele, kus kõik sõnad on üksteisest tühikutega eraldatud.

NB! Lehe pöördel on veel üks ülesanne!

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.

2. Riigikogu stenogramm (13p) ylesanne2.py

Failis *stenogramm1.txt* (kodeering UTF-8) on Riigikogu istungi stenogramm. Iga sõnavõtu ees on tühi rida, sellele järgneb eraldi real tekst "SÕNAVÕTT: " ja sõnavõtja nimi (mõne inimese puhul koos ametinimetusega) ning seejärel ühel või enamal järjestikusel real sõnavõtu tekst. Sõnavõtu lõpus on alati tühi rida.

Alamülesanne 1. Kirjuta funktsioon `sõnavõtja_nimi`, mis võtab argumendiks suvalise sõne. Kui tegemist on sõnavõtu alguse reaga, siis peab funktsioon tagastama sõnavõtja nime, vastasel juhul tuleb tagastada tühi sõne. Ametinimetuse eemaldamise lihtsustamiseks eeldame praegu, et igal inimesel on üks eesnimi ja üks perenimi.

Näited:

```
>>> sõnavõtja_nimi("V A H E A E G\n")
''
>>> sõnavõtja_nimi("SÕNAVÕTT: Peep Aru\n")
'Peep Aru'
>>> sõnavõtja_nimi("SÕNAVÕTT: Aseesimees Jüri Ratas\n")
'Jüri Ratas'
>>> sõnavõtja_nimi("Majandus- ja kommunikatsiooniminister Juhan Parts\n")
''
>>> sõnavõtja_nimi("SÕNAVÕTT: Majandus- ja kommunikatsiooniminister Juhan Parts\n")
'Juhan Parts'
```

Alamülesanne 2. Programmi põhiosa ülesanne on leida kõige pikem sõnavõtt ning väljastada ekraanile vastava sõnavõtja nimi. Seejuures tuleks sobivalt kasutada esimeses alamülesandes kirjeldatud funktsiooni. Pikimaks sõnavõtuks loeme seda, mille tekstis on kõige rohkem sümboleid. Kui failis on mitu pikimat sõnavõttu, siis tuleb tagastada neist viimane.

Programmi kasutuse näited (kasutaja sisend on kursiivis):

```
Sisesta failinimi: stenogramm1.txt
Juhan Parts
```

```
Sisesta failinimi: stenogramm2.txt
Marika Tuus-Laul
```

```
Sisesta failinimi: stenogramm3.txt
Priit Toobal
```

Vihje 1:

Abi on kahekordsest tsüklis. Kui välimine tsükel leiab failist sõnavõtja rea, siis lastakse sisemisel tsüklil lugeda kokku sõnavõtu tekst.

Vihje 2:

```
>>> "A".isalpha()
True
>>> "a".isalpha()
True
>>> "õ".isalpha()
True
>>> ":".isalpha()
False
>>> "3".isalpha()
False
```

Lihtsustus (-3p):

Kui jääd hätta mitmerealiste sõnavõttude arvestamisega, siis arvesta ainult sõnavõtu esimest rida. Testimiseks sobib siis fail *stenogramm3.txt*, sest seal on kõik sõnavõttud üherealised.

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.