

L06 Tehtävät

- Tekstitiedoston käsittely eli lukeminen ja kirjoittaminen
- Merkkijonomuuttujiin liittyvät jäsenfunktiot ja merkkijonojen muokkaus
- Usean tiedoston käsittely rinnakkainen
- Tiedostossa olevan numerotiedon analyysi: summa, minimi ja maksimi
- Laajeneva valikkopohjainen ohjelma T5
- Käytä paaohjelma()-rakennetta kaikissa tehtävissä L05 mukaisesti

Lue oppaan tämän viikon asioita käsittelevä luku 6. Lisäksi tehtävien suorittamiseen tarvitset aiempien lukujen tietoja. Ohjelmointitehtävät palautetaan Moodlen kautta CodeGradeen.

L06T1: Tekstitiedoston kirjoittaminen ja lukeminen	2
L06T2: Tiedoston rivimäärän ja merkkien laskeminen	2
L06T3: Tiedoston tietojen testaaminen, palindromit	3
L06T4: Tekstitiedoston tietojen tilastollinen analysointi	3
L06T5: Valikkopohjainen ohjelma / laskin, jatkoa tehtävälle T05T5	4

Huomioita ohjelmointitehtävistä, joissa käsitellään tiedostoja:

- Mikäli viikkotehtävissä tarvitaan valmiita tiedostoja, ovat ne kaikki Moodlessa viikon tehtävien jälkeen yhdessä hakemistossa.
- Käytä tiedostojen tarkasteluun koodieditoria, esim. IDLEä. Windowsin **Notepad ei näytä rivinvaihtoja oikein** ja tekstinkäsittelyohjelma kuten MS-Word ei sovi myöskään tähän tarkoitukseen.
- IDLE näyttää oletusarvoisesti Python tiedostoja ja esim. tekstitiedostot pitää valita erikseen näkyville, ts. avatessasi tiedostoa (File | Open), valitse tiedostotyyppiksi *.txt tai *.*.
- Alla on kunkin tehtävänannon kohdalla on merkitty käsiteltävät tiedostot **keltaisella**.
- **Mikäli tehtävässä luetaan tiedostoa**, tulee luettavan tiedoston olla **samassa hakemistossa Python-koodin kanssa** (mieluiten tehtäviä varten omalle kovalevyillesi tekemäsi hakemisto).
- Jos ohjelma luo tiedostoja, tutki resurssienhallinnan ja editorin avulla, mitä ja minkälaisia tiedostoja ohjelmasi teki ajon aikana. Muuten et voi löytää virheitä koodistasi. Tutki myös luettavien tiedostojen rakenne tarkasti ennen tehtävän aloittamista. Ja tiedostoja avatessasi käytä IDLEä...

L06T1: Tekstitiedoston kirjoittaminen ja lukeminen

Harjoitellaan tiedoston kirjoittamista ja lukemista aliohjelmissa.

1. Tee kolmesta aliohjelmasta muodostuva ohjelma luentokalvojen valikkopohjaisen ohjelman tyyliin. Nyt tarvittavat ohjelmat ovat paaohjelma, TiedostoKirjoita ja TiedostoLue. Kysy pääohjelmassa tallennettavan tiedoston nimi ja välitä se parametrina molempiin aliohjelmiin.
2. Tiedon kirjoitus -aliohjelman tulee kysyä käyttäjältä nimi ja tallentaa sen tekstitiedostoon. Kysy nimi toistorakenteessa ja lopeta kysyminen, kun käyttäjä antaa syötteeksi 0:n (ts. numerosta nolla muodostuva merkkijono). Tiedosto tulee avata aliohjelman alussa ja mahdollinen aiempi tiedoston sisältö tulee tuhota avauksen yhteydessä. Aliohjelman lopuksi tiedosto tulee sulkea.
3. Tiedoston luku -aliohjelmalla varmistetaan, että tiedostossa on haluttu sisältö. Avaa tiedosto, lue sen sisältö ja tulosta näytölle yksi nimi riville sekä lopuksi sulje tiedosto.

Huomaa, että tässä tehtävässä pitää toteuttaa oikein toistorakenne (L04), aliohjelmat (L05) ja tiedostonkäsittely (L06). Virhe missä tahansa näistä kohdista estää ohjelman oikean toiminnan, joten tarkista em. rakenteet aiemmilta viikoilta tarpeen mukaan.

Ohjelman esimerkkiajo:

```
Anna tallennettavan tiedoston nimi: L06T1T1.txt
Anna tiedostoon tallennettava nimi (0 lopettaa): Ville
Anna tiedostoon tallennettava nimi (0 lopettaa): Kalle
Anna tiedostoon tallennettava nimi (0 lopettaa): Joonatan
Anna tiedostoon tallennettava nimi (0 lopettaa): 0
Tiedostoon 'L06T1T1.txt' on tallennettu seuraavat nimet:
Ville
Kalle
Joonatan
Kiitos ohjelman käytöstä.
```

L06T2: Tiedoston rivimäärän ja merkkien laskeminen

Tee ohjelma, joka lukee tekstitiedoston ja laskee siinä olevien rivien sekä merkkien määrät. Tiedosto **L06T2D2.txt** sisältää kaikki väestörekisterissä elokuussa 2020 olleet naisten nimet, joita oli vähintään viidellä ihmisellä (tietosuojasyiden vuoksi tätä harvinaisempia nimiä ei julkaista). Laskemalla rivien määrän saat selville, kuinka monta nimeä tiedostossa on. Laske lisäksi tiedostossa olevien merkkien määrä sekä nimien keskimääräinen pituus alla olevan esimerkkiajon mukaisesti. Tiedostossa **L06T2D3.txt** on vastaavasti kaikkien miesten nimet. Ohjelman toiminta ja laskenta kannattaa testata ensin toimivaksi tiedostolla **L06T2D1.txt**, jossa on nimet Tapio ja Matti, jotta voit helposti varmistua laskennan oikeellisuudesta. Käytä ohjelmassa paaohjelma()-rakennetta ja avaa tiedosto UTF-8 koodattuna, jotteivat ääkköset sotke asioita. Tiedostossa olevien merkkien määrää laskiessa kannattaa tarvittaessa tarkistaa, mikä on rivin ja nimen pituus.

Ohjelman esimerkkiajo:

```
Anna luettavan tiedoston nimi: L06T2D2.txt
Tiedostossa oli 12356 nimeä ja 101032 merkkiä.
Keskimääräinen nimen pituus oli 7 merkkiä.
Kiitos ohjelman käytöstä.
```

L06T3: Tiedoston tietojen testaaminen, palindromit

Tämä tehtävä jatkaa aiempaa tehtävää L03T4. Tällä kertaa testattavat merkkijonot luetaan tiedostosta ja palindromit kirjoitetaan toiseen tiedostoon. Toteuta ohjelma yhtenä pääohjelmana.

Kysy pääohjelman alussa luettavan tiedoston nimi ja määritä kirjoitettavan tiedoston nimeksi **L06T3T1.txt**. Avaa molemmat tiedostot, lue tiedostosta yksi rivi kerrallaan ja luokittele se. Tiedostossa voi olla kahdenlaisia rivejä: numerorivejä, jotka sisältävät pelkkiä numeroita ja jotka hylätään, tai kirjainrivejä, jolloin niille tehdään palindromitestaus ja palindromit kirjoitetaan tiedostoon. Ohjelmasi tulee tulostaa jokaisen rivin sisältö ja testauksen tulos eli "on palindromi", "ei ole palindromi" tai "on numerorivi" alla olevan esimerkkiajon mukaisesti.

Varsinainen testiaineisto on tiedostossa **L06T3D1.txt**, joka sisältää numerorivejä ja merkkirivejä, merkkiriveistä osa on palindromeja ja osa ei ole. Ohjelman eri toimintojen testaamisen helpottamiseksi Moodlessa on kaksi muuta tiedostoa, joista tiedosto **L06T3D2.txt** sisältää kaksi palindromia ja yhden merkkijonon, joka ei ole palindromi. Tiedostossa **L06T3D3.txt** on kaksi numeroriviä eikä muuta. Näillä tiedostoilla voi testata rivien oikean luokittelun ennen varsinaisen testitiedoston käsittelyä.

Numeroiden testauksen voi tehdä ohjelmointioppaan mukaisesti isdigit()-jäsenfunktiolla. Kannattaa muistaa, että readline()-funktio palauttaa merkkijonossa myös rivinvaihtomerkin, eli "apina" luetaan tiedostosta merkkijonona "apina\n". Rivinvaihtomerkkiä ei tule huomioida numero-tarkastelussa eikä palindromitestauksessa ja vain rivinvaihtomerkin sisältävä rivi lopettaa tiedoston lukemisen. Luettava tiedosto on UTF-8 koodattu.

Ohjelman esimerkkiajo:

```
Anna luettavan tiedoston nimi: L06T3D1.txt
Rivi 'attekumioravavaroimuketta' on palindromi.
Rivi '234242' on numerorivi.
Rivi 'isotupakka' ei ole palindromi.
Rivi '0234234' on numerorivi.
Rivi 'eiolepalindromi' ei ole palindromi.
Rivi 'nallevaiviiviavellan' on palindromi.
Rivi 'eitamakaan' ei ole palindromi.
Rivi 'innostunutsonni' on palindromi.
Rivi '123909' on numerorivi.
Rivi 'emmanamme' on palindromi.
Rivi 'aatuosaasoutaa' on palindromi.
Rivi '34422424' on numerorivi.
Rivi 'diipadaapa' ei ole palindromi.
Kiitos ohjelman käytöstä.
```

L06T4: Tekstitiedoston tietojen tilastollinen analysointi

Tee ohjelma, joka selvittää vuoden päivittäiset askelmäärät sisältävästä tekstitiedostosta koko vuoden askelmäärän ja pienimmän sekä suurimman päivittäisen askelmäärän. Tiedosto sisältää yhdellä rivillä aina yhden päivän askeleet, joten koko vuoden askelmäärä selviää laskemalla kaikki tiedostossa olevat numerot yhteen. Pienin ja suurin askelmäärä selviää vertailemalla aina kahta askelmäärää kerrallaan ja ottamalla niistä pienemmän/suuremman mukaan jatkovertailuun, kunnes kaikkia luvut on käyty läpi ja pienin/suurin määrä löytynyt.

Toteuta ohjelma pääohjelman ympärille, jossa kysytään ensin luettavan tiedoston nimi ja sen jälkeen kutsutaan kolmea aliohjelmaa, joista kukin selvittää yhden asian: askelien yhteismäärän, pienimmän askelmäärän ja suurimman askelmäärän. Näiden tietojen

selvittämisen jälkeen kysy pääohjelmassa tallennettavan tiedoston nimi ja lähetä se sekä selvitettyt tiedot aliohjelmaan, joka tulostaa ja tallentaa samat merkkijonot esimerkkiajon mukaisesti.

Vinkkejä ohjelman toteutukseen: Huomaa, että kaikissa tiedostoa analysoivissa aliohjelmissa on sama runko – avaa tiedosto, lue yksi rivi ja analysoi se, että saat koko tiedostoa koskevan tuloksen selville ja palauta se kutsuvaan ohjelmaan. Siksi kannattaa tehdä ensin yksi aliohjelma, joka selvittää yhden asian ja kun se onnistuu, kopioi aliohjelma toiseksi aliohjelmaksi ja muuta sen analyysiä niin, että se selvittää toisen asian jne. Useiden lukujen summan laskenta on käyty läpi mm. luennoilla keskiarvon laskennan yhteydessä, joten tarvittaessa saat apua algoritmiin aiemmista luentokalvoista. Pienimmän ja suurimman voi selvittää esimerkiksi vertailemalla kahta tiedostossa olevaa lukua toisiinsa ja jatkamalla vertailua niin pitkään, että haluttu pienin/suurin arvo löytyy kaikkien lukujen vertailun jälkeen. Tällainen vertailu kannattaa aloittaa alustamalla haluttu apumuuttuja, esim. pienin, ensimmäisellä tiedostossa olevalla arvolla ja vertaamalla sitä sitten edellä olevan mukaisesti muihin lukuihin.

Ohjelman lukemat tiedot löytyvät tiedostosta **L06T4D1.txt**, jossa on päivittäinen askelmäärä eli yksi kokonaisluku per rivi ja rivi jokaiselle vuoden päivälle. Moodlesta on myös tiedosto **L06T4D2.txt**, jossa on kaksi lukua, joiden avulla voit helposti varmistua ohjelmasi oikeasti toiminnasta. Tallennettava tiedosto **L06T4T1.txt** löytyy myös Moodlesta, jotta voit tarvittaessa varmistua sen muodosta. Tässä tehtävässä voidaan olettaa, että analysoitavassa tiedostossa on vähintään yksi rivi.

Ohjelman esimerkkiajo:

Anna tiedot sisältävän tiedoston nimi: L06T4D1.txt
Anna tallennettavan tiedoston nimi: L06T4T1.txt
Pienin askelmäärä oli 1723 askelta.
Suurin askelmäärä oli 21662 askelta.
Yhteensä askelia oli 3451029 askelta.
Kiitos ohjelman käytöstä.

Luettavan tiedoston muoto, L06T4D2.txt:

10000
12345

Kirjoitettavan tiedoston muoto, L06T4T2.txt:

Pienin askelmäärä oli 10000 askelta.
Suurin askelmäärä oli 12345 askelta.
Yhteensä askelia oli 22345 askelta.

L06T5: Valikkopohjainen ohjelma / laskin, jatkoa tehtävälle T05T5

Tämä tehtävä laajentaa aiemmin tehtyä laskinta, jossa valikkopohjaisen laskimen valintarakenne tehtiin tehtävässä L03T3, toistorakenne L04T5 ja aliohjelmarakenne L05T5. Lisää aliohjelmissa muodostuvaan versioon tiedostonkäsittely eli lue syötteet yhdestä tiedostosta ja kirjoita tulokset toiseen tiedostoon. Tämän tehtävän lähtökohtana kannattaa käyttää itse tekemääsi ohjelmaa L05T5.

Tällä kertaa tarvitsee muuttaa vain pääohjelmaa ja lukujen kysymisen hoitavaa aliohjelmaa. Lisää pääohjelmaan luettavan tiedoston nimen kysyminen ja avaa se lukemista varten. Tämän jälkeen avaa tulosten kirjoittamista varten tiedosto **L06T5T1.txt**. Kun käyttäjä valitsee

lukuja antamisen, tulee pääohjelman kutsua LueLuku-aliohjelmaa tiedostokahva parametrina ja tämä aliohjelma palauttaa seuraavalta riviltä luetun luvun kokonaislukuna. Kutsu samaa aliohjelmaa toisen kerran, jolloin saat toisen luvun ohjelman laskentaa varten ja tulosta ne esimerkkiajon mukaisesti näytölle. Mikäli tiedostossa ei ole enää lukuja, tulostaa aliohjelma näytölle viestin "Luvut loppuivat, lopeta ohjelma." ja palauttaa arvon 0 esimerkkiajon mukaisesti. Onnistuneen laskennan jälkeen ohjelma kirjoittaa laskenta-aliohjelman palauttaman merkkijonon tiedostoon uudelle riville ja tulostaa näytölle tiedon tallennuksesta. Käyttäjän valittua lopetuksen ohjelma sulkee molemmat tiedostot ja kiittää ohjelman käytöstä esimerkkiajon mukaisesti.

Moodlessa on tiedostot **L06T5D1.txt** ja **L06T5D2.txt**, joissa on erilaiset syötteet ohjelmalle ja ohjelman esimerkkitalenne **L06T5T1.txt**.

Luettavan tiedoston muoto, **L06T5D2.txt:**

1
2

Kirjoitettavan tiedoston muoto, **L06T5T1.txt:**

Summa $2 + 8 = 10$

Osamäärä $1 / 3 = 0.33$

Summa $2 + 2 = 4$

Ohjelman esimerkkiajo:

Anna luettavan tiedoston nimi: L06T5D2.txt

Tämä laskin osaa seuraavat toiminnot:

- 1) Anna luvut
- 2) Summa
- 3) Osamäärä
- 0) Lopeta

Valitse toiminto (0-3): 1

Luettiin luvut 1 ja 2

Tämä laskin osaa seuraavat toiminnot:

- 1) Anna luvut
- 2) Summa
- 3) Osamäärä
- 0) Lopeta

Valitse toiminto (0-3): 2

Tulos tallennettu tiedostoon.

Tämä laskin osaa seuraavat toiminnot:

- 1) Anna luvut
- 2) Summa
- 3) Osamäärä
- 0) Lopeta

Valitse toiminto (0-3): 1

Luvut loppuivat, lopeta ohjelma.

Luvut loppuivat, lopeta ohjelma.

Luettiin luvut 0 ja 0

Tämä laskin osaa seuraavat toiminnot:

- 1) Anna luvut
- 2) Summa
- 3) Osamäärä
- 0) Lopeta

Valitse toiminto (0-3): 0

Lopetetaan

Kiitos ohjelman käytöstä.