

L12 Tehtävät

- Henkilötunnuksen tarkastusmerkin käsittely, binäärilukujen laskutoimitus, poikkeusten käsittely, tekstitiedostojen CSV ja JSON-muodot, omatoiminen ongelmanratkaisu

Lue oppaan tämän viikon asioita käsittelevä luku 12 ja lisäksi tehtävien suorittamiseen tarvittavat aiempien lukujen tiedot. Ohjelmointitehtävissä on oltava otsikkotiedot ja ne palautetaan CodeGradeen.

L12T1: Henkilötunnuksen oikeellisuuden tarkastaminen	1
L12T2: Binaarilukujen erotus	1
L12T3: Valikkopohjainen ohjelma CSV ja JSON tiedostojen käsittelyyn.....	2

L12T1: Henkilötunnuksen oikeellisuuden tarkastaminen

Henkilötunnus on esimerkki koodatusta tiedosta, kuten luentojen 2 ja 3 ohjelmointivideoilla oli puhetta. Tee ohjelma, joka testaa onko käyttäjän antama henkilötunnus muodostettu oikein. Wikipedian artikkeli <http://fi.wikipedia.org/wiki/Henkilötunnus> kertoo, kuinka henkilötunnus muodostetaan ja sinun tulee toteuttaa tunnuksen tarkastus automaattisesti tietokoneohjelman avulla. Ohjelman tulee hyväksyä sekä isot että pienet kirjaimet syötteessä.

Ohjelman esimerkkiajo 1:

Anna henkilötunnus: 120345-678M
Henkilötunnus hyväksytty.
Kiitos ohjelman käytöstä.

Ohjelman esimerkkiajo 2:

Anna henkilötunnus: 123456-4444
Henkilötunnusta ei hyväksytä.
Kiitos ohjelman käytöstä.

L12T2: Binaarilukujen erotus

Tee ohjelma, joka pyytää käyttäjältä kaksi binaarilukua ja laskee niiden erotuksen.

Huomaa, että luku on luku eli numero sen esitystavasta riippumatta ja kantaluku on esitystapaan liittyvä asia. Binaariluku tarkoittaa sitä, että luvun esitysmuodossa on vain ykkösiä ja nollia, ja siksi binaariluku on helpointa syöttää tietokoneohjelmalle merkkijonona, jossa on ykkösiä ja nollia. Itse laskutehtävän voi ratkaista esim. muuttamalla binaariluvut ensin kymmenkantaisiksi luvuiksi ja suorittamalla laskun niillä.

Tässä tehtävässä binaariluvun muunnos 10-kantaiseksi kokonaisluvuksi tulee toteuttaa itse omalla algoritmilla eikä valmiita ratkaisuja saa käyttää. Samoin tässä tehtävässä keskitytään vain kokonaislukuihin eikä desimaalilukuja tarvitse huomioida. Lisätietoa binaarijärjestelmästä löydät luentokalvoista ja Wikipediasta <http://fi.wikipedia.org/wiki/Binäarijärjestelmä>.

Ohjelman esimerkkiajo:

```
Anna ensimmäinen binaariluku: 100100001
Anna toinen binaariluku: 10011101
Bittijonosi 100100001 on kymmenkantaisena kokonaislukuna 289
Bittijonosi 10011101 on kymmenkantaisena kokonaislukuna 157
Lukujen 289 ja 157 erotus on 132
Kiitos ohjelman käytöstä.
```

L12T3: Valikkopohjainen ohjelma CSV ja JSON tiedostojen käsittelyyn

Tämä ohjelma käsittelee kahdessa eri muodossa olevia tekstitiedostoja. Kurssilla on käytetty usein CSV-tiedostoja eli *Comma Separated Values* -tiedostoja, joissa rivillä on pilkulla erotettuja arvoja. Taulukkolaskentaohjelmat tukevat tällaisten tiedostojen lukemista ja kirjoittamista ja siksi näiden tiedostojen päätte on tyypillisesti .CSV. Toinen tyypillinen tekstitiedostoformaatti on JSON, jossa on tarkemmin määritelty rakenne siten, että jokaisen tietoalkion yhteydessä on sitä vastaavan muuttujan nimi. Nämä molemmat tiedostotyytit on selitetty tarkemmin esim. Wikipediassa, https://en.wikipedia.org/wiki/Comma-separated_values ja <https://en.wikipedia.org/wiki/JSON>, ja tässä tehtävässä on esimerkit molemmista tiedostoformaateista sekä niiden välillä tarvittava muunnos.

Tee ohjelma, jolla voi lukea ja kirjoittaa CSV ja JSON tiedostoja. CSV-tiedoston luku ja kirjoitus tulee tehdä samalla tavalla kuin aiemminkin tällä kurssilla eli readline ja write -jäsenfunktioilla, mutta JSON-tiedoston käsittely tulee tehdä jsonpickle-kirjaston avulla ohjelmointivideon esimerkin mukaisesti. Käyttäjä voi lukea CSV-tiedoston ja kirjoittaa sen JSON-muodossa, tai lukea JSON-tiedoston ja kirjoittaa sen CSV-muodossa esimerkkiajon valikon mukaisesti. CSV-tiedoston sarakkeiden nimet kertovat mitä tietoa sarakkeessa on, joten niistä saa tässä tehtävässä käytettävän luokan jäsenmuuttujien nimet.

Tiedostot L12T3D1.csv ja L12T3D2.csv sisältävät HelMet kirjastojen varatuimpien kirjojen tietoja CSV-muodossa. Tiedostot L12T3D1.json ja L12T3D2.json sisältävät samat tiedot JSON-muodossa, D1 tiedosto on pienempi testitiedosto ja D2 sisältää laajemman datasetin. Alkuperäinen data löytyy osoitteesta <https://www.avoindata.fi/data/fi/dataset/helmet-kirjastojen-varatuimmat-kirjat>. Dataa on muokattu niin, että Moodlen CSV tiedostoissa kentät on erotettu puolipisteellä ja koodauksessa käytetään UTF-8 merkkistöä. JSON-tiedostot perustuvat myös UTF-8 koodaukseen, mutta ääkköset ovat ASCII-merkkeinä.

CSV-data sisältää seitsemän tietoalkiota per rivi ja ohjelmassasi on oltava luokka TIEDOT, jossa on näitä sarakkeita vastaavat jäsenmuuttujat. JSON-tiedostossa näkyy luokan nimi ja kaikkien jäsenmuuttujien nimet, joten ne kannattaa katsoa sieltä. Jäsenmuuttujat Nimike, Tekijä ja ISBN tietoalkiot tulee käsitellä merkkijonoina. Varauksia / Nide tietoalkio on liukuluku ja loput ovat kokonaislukuja. Tässä tehtävässä voidaan lisäksi olettaa, ettei tietoalkioissa ole ylimääräisiä puolipisteitä eikä keskellä tiedostoa ole tyhjiä rivejä. Jokaisella rivillä olevat tiedot tulee sijoittaa yhden olion jäsenmuuttujiin ja oliot lisätään oliolistaan. Huomaa, että "varauksia/nide" -tieto on desimaaliluku ja CSV-tiedostossa saattaa siten olla desimaalipilkku. Tämän voi muuttaa pisteeksi esim. replace-jäsenfunktioilla ennen merkkijonon muuttamista desimaaliluvuksi.

Oliot tulee muuttaa JSON muotoon ja takaisin jsonpickle-kirjastolla. Tämä kirjasto ei kuulu Pythonin standardikirjastoon, mutta sen voi asentaa numpy-kirjaston tavoin eli käskyllä **pip install jsonpickle** (ks. L08). Kirjaston dokumentaatio on osoitteessa <https://jsonpickle.readthedocs.io/en/latest/index.html>. Tässä tehtävässä kannattaa tutustua encode() ja decode() -jäsenfunktioihin. encode()-jäsenfunktio saa parametrina Python olion ja

palauttaa sen JSON muotoisena merkkijonona. Tässä tehtävässä tulee käyttää encode()-jäsenfunktion parametria indent ja antaa sille arvoksi 4, jotta JSON-tiedoston rakenne on luettavissa helpolla. decode()-jäsenfunktio saa parametrinä JSON-dataa sisältävän merkkijonon ja palauttaa Python rakenteen (lista, olio, oliolista tms.). Funktioiden muita parametreja ei tarvita tässä tehtävässä. Kannattaa huomata, että encode() muuttaa ääkköset ASCII merkeiksi (esim. ä = \u00e4) ja decode() muuttaa ASCII ääkköset takaisin UTF-8 muotoon. Lisäksi JSON-muotoinen data tulee lukea yhdellä käskyllä, jotta JSON-CSV -muunnos toimii oikein.

Toteuta ohjelman toiminnallisuudet omissa aliohjelmissaan, esim. tiedoston nimen kysymiseen riittää yksi funktio, jolle laitetaan sopiva merkkijono parametrinä. Näin saat ohjelmalle selkeän rakenteen ja tiedostonkäsittelyihin normaalit virheenkäsittelyt. Kerro käyttäjälle ohjelman etenemisestä eli tulosta pääohjelmassa tilannetietona aina suoritettu aliohjelma ja siihen liittyvä lukumäärätieto esimerkkiajon mukaisesti.

Esimerkki CSV-tiedostosta:

```
Nimike;Tekijä;ISBN;Varauksia;Niteitä;Tilattuja lisäkappaleita;Varauksia / nide
Tyttö, sinä olet... ;Pääskysaari, Jenni ;9789511202851;145;56;0;2.6
Arvoitus nimeltä Margo ;Green, John ;9789510411759;145;121;0;1.2
Ufoja Lahdessa : kokoelma ;Turunen, Marko ;9789525915167;6;2;0;3.0
Vapaa kuin lintu : Emil Nervanderin elämä ;Valkeapää, Leena
;9789525533200;6;2;0;3.0
Urheilijan Taivas+helvetti : paras versio itsestäsi ;Hämäläinen, Karo
;9789526794693;144;15;18;4.4
Ihminen : fysiologia ja anatomia ;;9789510346556;139;64;1;2.1
The one thing : the surprisingly simple truth behind extraordinary results ;Keller,
Gary ;9781848549258;18;6;0;3.0
Blod på snö ;Nesbø, Jo ;9789164204349;16;15;0;1.1
Reaktio : lukion kemia. 3 : Reaktiot ja energia ;;9789526303239;18;2;0;9.0
Girlboss ;Amoruso, Sophia ;9780241217931;18;1;2;6.0
```

Esimerkki JSON-tiedoston alusta:

```
[
  {
    "py/object": "__main__.TIEDOT",
    "Nimike": "Tyttö\u00f6, sin\u00e4 olet... ",
    "Tekija": "P\u00e4\u00e4skysaari, Jenni ",
    "ISBN": "9789511202851",
    "Varauksia": 145,
    "Niteita": 56,
    "Lisakappaleita": 0,
    "VarauksiaPerNide": 2.6
  },
  {
    "py/object": "__main__.TIEDOT",
    "Nimike": "Arvoitus nimelt\u00e4 Margo ",
    ...
  }
]
```

Tehtävän L12T3 ohjelman esimerkkiajo:

Valitse haluamasi toiminto:

- 1) Lue CSV tiedosto
- 2) Lue JSON tiedosto
- 3) Kirjoita CSV tiedosto
- 4) Kirjoita JSON tiedosto
- 0) Lopeta

Anna valintasi: 1

Anna luettavan CSV tiedoston nimi: L12T3D1.csv

Luettu 10 kirjan tiedot.

Valitse haluamasi toiminto:

- 1) Lue CSV tiedosto
- 2) Lue JSON tiedosto
- 3) Kirjoita CSV tiedosto
- 4) Kirjoita JSON tiedosto
- 0) Lopeta

Anna valintasi: 4

Anna kirjoitettavan JSON tiedoston nimi: L12T3D1.json

Tiedosto L12T3D1.json kirjoitettu.

Valitse haluamasi toiminto:

- 1) Lue CSV tiedosto
- 2) Lue JSON tiedosto
- 3) Kirjoita CSV tiedosto
- 4) Kirjoita JSON tiedosto
- 0) Lopeta

Anna valintasi: 2

Anna luettavan JSON tiedoston nimi: L12T3D1.json

Luettu 10 kirjan tiedot.

Valitse haluamasi toiminto:

- 1) Lue CSV tiedosto
- 2) Lue JSON tiedosto
- 3) Kirjoita CSV tiedosto
- 4) Kirjoita JSON tiedosto
- 0) Lopeta

Anna valintasi: 3

Anna kirjoitettavan CSV tiedoston nimi: L12T3T1.csv

Tiedosto L12T3T1.csv kirjoitettu.

Valitse haluamasi toiminto:

- 1) Lue CSV tiedosto
- 2) Lue JSON tiedosto
- 3) Kirjoita CSV tiedosto
- 4) Kirjoita JSON tiedosto
- 0) Lopeta

Anna valintasi: 0

Kiitos ohjelman käytöstä.