

L12 Tehtävät

- Henkilötunnuksen tarkastusmerkin käsittely, binäärilukujen laskutoimitus, poikkeusten käsittely, tiedoston siistiminen, 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 Moodlen kautta CodeGradeen.

L12T1: Henkilötunnuksen oikeellisuuden tarkastaminen	1
L12T2: Binaarilukujen erotus	2
L12T3: Tekstitiedoston ja merkkijonojen käsittely	3

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 kirjastoista löytyviä ratkaisuja saa käyttää. Samoin tässä tehtävässä keskitytään vain kokonaislukuihin eikä desimaalilukuja tarvitse huomioida. Lisätietoa binaarijärjestelmästä löydät luentokalvoista ja Wikipediasta <https://fi.wikipedia.org/wiki/Binäarijärjestelmä>.

Ohjelman esimerkkiajo:

```
Anna ensimmäinen binaariluku: 10100001
Anna toinen binaariluku: 10011101
Bittijonosi 10100001 on kymmenkantaisena kokonaislukuna 161
Bittijonosi 10011101 on kymmenkantaisena kokonaislukuna 157
Lukujen 161 ja 157 erotus on 4
Kiitos ohjelman käytöstä.
```

L12T3: Tekstitiedoston ja merkkijonojen käsittely

Tee ohjelma, joka lukee tiedoston, hylkää epäkelvot rivit ja moukkaa hyväksyttävissä olevat rivit oikeaan muotoon sekä kirjoittaa ne uuteen tiedostoon.

Tiedostot L12T3D1.txt ja L12T3D2.txt sisältävät dataa, joiden riveillä on puolipisteillä erotettuna kolme tietoalkiota eli kommentoijan *ID*, *kommentti* ja *mainepisteet*. Ohjelma tarkistaa ensin, että rivillä on oikea määrä tietoalkiota ja että ID on hyväksyttävä. Mikäli näin ei ole, rivi hylätään ja tulostetaan hylkäyksestä kertova virheilmoitus ko. rivin kanssa (ks. esimerkkiajo). Kaikki hyväksyttävät rivit kirjoitetaan uuteen tiedostoon mahdollisten muokkausten jälkeen, tiedoston nimi kysytään käyttäjältä.

Tietoalkioihin liittyvät vaatimukset ovat seuraavat. *ID* on nelinumeroinen luku, esimerkiksi 0023, ja luvussa voi olla etunollia esimerkin mukaisesti. *Kommentti*-kentässä on kommentteja, jotka ovat satunnaisia alfanumeerisia merkkejä eli kirjaimia ja numeroita sisältäen mahdollisesti myös muita eli virheellisiä merkkejä. Poista *kommentti*-kentästä kaikki muut paitsi alfanumeeriset merkit, jolloin esimerkiksi *kommentista* "par!si-ttu 55" tulee "parsittu55". *Mainepisteiden* tulee olla kokonaisluku ja mikäli se puuttuu tai on jotain muuta, korvataan se arvolla 0. Tässä tehtävässä voidaan olettaa, ettei kommentteissa ole puolipisteitä eli kenttien erotinmerkkejä eikä rivinvaihtomerkkejä eikä keskellä tiedostoa ole tyhjiä rivejä.

Toteuta ohjelma kolmena aliohjelmana, joista ensimmäinen lukee tiedoston, toinen analysoi tiedot ja kolmas kirjoittaa tiedoston. Näin saat ohjelmalle selkeän rakenteen ja tiedostonkäsittelyihin normaalit virheenkäsittelyt. Analyysiosiossa kannattaa miettiä, miten tietoalkiot on helpointa tarkistaa ja korjata tarpeen mukaan. Esim. kommentin virheelliset merkit voi hoitaa kopioimalla hyväksyttävät merkit toiseen merkkijonoon ja jättämällä ei-hyväksyttävät merkit kopioimatta sinne. Mainepisteiden tarkistus onnistuu esim. poikkeustenkäsittelyllä muutettaessa merkkijono kokonaisluvuksi.

Kerro käyttäjälle ohjelman etenemisestä eli tulosta pääohjelmassa tilannetietona aina suoritettu aliohjelma ja siihen liittyvä lukumäärätieto.

Esimerkki luettavan tiedoston alusta:

```
ID;Kommentti;Mainepisteet
0001;mvseibmosm5;100
0002;s5semh5ehse5;-42
0003;spmss4easm4445;
```

Ohjelman esimerkkiajo 1:

```
Anna luettavan tiedoston nimi: L12T3D1.txt
Anna kirjoitettavan tiedoston nimi: L12T3T1.txt
Tiedosto 'L12T3D1.txt' luettu, 17 riviä.
Virheellinen ID, rivi 12: ';!fmbp-df;b'
Virheellinen ID, rivi 13: ';;'
Väärä määrä arvoja, rivi 14: ';'
Väärä määrä arvoja, rivi 15: ';;;
Virheellinen ID, rivi 16: '99999;flipflop!!;123'
Tiedot analysoitu, 11 hyväksyttävää tietoalkiota.
Tiedosto 'L12T3T1.txt' kirjoitettu, 12 riviä.
Kiitos ohjelman käytöstä.
```

Ohjelman esimerkkiajo 2:

```
Anna luettavan tiedoston nimi: eiote.txt
Tiedoston 'eiote.txt' käsittelyssä virhe, lopetetaan.
```