

# Harjoitustyön perustason tehtävänanto

1. Tehtävä.....	1
2. Ohjelman vaatimukset .....	1
3. Luettavan tiedoston rakenne .....	2
4. Tulostettavan ja kirjoitettavan tiedoston rakenne .....	2
5. Esimerkkiajo .....	3
6. Virheilmoitukset .....	3

## 1. Tehtävä

Tee C-kielinen ohjelma, joka lukee tekstitiedoston ja luo sen riveillä olevista tiedoista linkitetyn listan, analysoi tiedot ja kirjoittaa tulokset sen jälkeen näytölle sekä tiedostoon. Lähtökohtana on Moodlesta otettu loki-tiedosto, jossa on ohjelmointikurssin tehtävien palautusajankohdat, joista sinun tulee selvittää perustietoja kuten eri tehtävien palautusmäärät. Palauta ohjelmasi Viopeen ja tarkista, että se menee testiajoista läpi. Tee ohjelman kääntämiseen kurssin ohjeiden mukainen Makefile ja palauta se Viopeen palauttamiesi tiedostojen kanssa Moodleen. Muista laittaa jokaiseen palautettavaan tiedostoon otsikkotiedot.

## 2. Ohjelman vaatimukset

Ohjelmassa tulee olla seuraavat kohdat sisältävä päävalikko:

- 1) Lue tiedosto
- 2) Analysoi tiedot
- 3) Tulosta tulokset
- 0) Lopeta

Näihin toimintoihin liittyvät seuraavat toiminnot:

1. **Lue tiedosto.** Ohjelma kysyy käyttäjältä luettavan tiedoston nimen, lukee tiedoston ja muodostaa yhdellä rivillä olevista tiedoista aina yhden solmun linkitettyyn listaan. Solmuna oleva tietue sisältää yhden palautusajan, päivämäärä ja kellonaika, tehtävän nimen merkkijonona ja tunnistenumeron sekä palauttajan tunnistenumeron. Palautusaika tulee tallettaa solmuun aika-tietueena (struct tm), tehtävän nimi merkkijonona, tehtävän ja käyttäjän tunnistenumerot kokonaislukuina. Tiedoston lukemisen jälkeen ohjelma ilmoittaa luettujen tietueiden lukumäärän sekä ensimmäisen ja viimeisen palautuksen päivämäärät.
2. **Analysoi tiedot.** Ohjelma käy linkitetyn listan läpi ja laskee eri tehtävien palautusmäärät. Erilaisia tehtäviä on 60, joten voit tehdä taulukon näiden tehtävien palautusmäärille ja käyttää tehtävän tunnistetta taulukon indeksin lähtökohtana. Taulukkoon kannattaa laittaa tietueita, joissa on aina yhden tehtävän palautusten lukumäärä ja tehtävän nimi merkkijonona. Tietueet kannattaa alustaa laittamalla lukumääräksi 0 ja tehtävän nimeksi ”Tyhjä”.
3. **Tulosta tulokset.** Ohjelma kysyy käyttäjältä haluaako nämä tulokset näytölle vai tiedostoon, ja tiedoston tapauksessa vielä tallennettavan tiedoston nimen. Tämän

jälkeen tulostetaan/tallennetaan taulukko, jossa on jokaisen tehtävän nimen perässä sen palautusmäärä puolipisteellä erotettuna.

4. **Lopeta.** Ohjelma tyhjentää listan ja lopettaa ohjelman suorittamisen.

Muita ohjelman toimintaan liittyviä vaatimuksia:

1. Tee aliohjelma, jolla kysyt käyttäjältä tiedoston nimen ja käytä sitä sekä luettavan että kirjoitettavan tiedoston nimen kysymiseen.
2. Jaa ohjelma kolmeen kooditiedostoon ja vastaaviin otsikkotiedostoihin. Laita pääohjelma ja valikko-aliohjelma paaohjelma.c -tiedostoon, tiedostonkäsittelyyn liittyvät aliohjelmat ali1.c-tiedostoon ja muut aliohjelmat ali2.c-tiedostoon. Jokaisen C-tiedoston parina on saman niminen otsikkotiedosto.
3. Ohjelman tulee sulkea tiedostot aina tiedostonkäsittelyn päätyttyä.
4. Tiedostoa kirjoitettaessa mahdollinen aiempi tiedosto tuhotaan avaamisen yhteydessä.
5. Luettaessa tiedostossa oleva data listaan, on lista tyhjennettävä ennen datan lisäämistä.
6. Tarkista analysointi- ja tallennusaliohjelmissa ensimmäisenä, että listassa on dataa ja jos näin ei ole, ilmoita asiasta käyttäjälle ja palaa kutsuvaan ohjelmaan.

### 3. Luettavan tiedoston rakenne

Tehtävien palautusdata on seuraavassa muodossa:

```
Palautusaika;TehtavaNimi;TehtavaID;KayttajaID
8/09/20, 10:46;L01T1;1;6607164
8/09/20, 10:47;L01T3;3;6607164
8/09/20, 10:47;L01T4;4;6607164
8/09/20, 10:47;L01T2;2;6607164
8/09/20, 10:48;L01T5;5;6607164
8/09/20, 17:50;L01T1;1;3209775
...
```

Tiedostossa on yhden kurssin tehtävien palautusajat aikajärjestyksessä. Data on esikäsiteltyä eli sitä on muokattu Moodlesta oton jälkeen, jotta sen käsittely olisi selkeämpää. Etukäteen ei voi tietää, onko kaikilta päiviltä ja kaikista tehtävistä palautuksia.

Moodlessa ja Viopessa on useampi testitiedosto. Ohjelman teko kannattaa aloittaa 15 riviä tietoja sisältävällä data15.csv -tiedostolla. Varsinainen analysoitava data on noin 7500 riviä sisältävässä tiedostossa data7500.csv, jossa tietoja on noin kuuden kuukauden ajalta. Erilaisia poikkeustilanteita voi testata tiedostolla data5800.csv.

### 4. Tulostettavan ja kirjoitettavan tiedoston rakenne

Ohjelman tulee tulostaa näytölle täsmälleen samat tiedot mitä se kirjoittaa tiedostoon.

Kirjoitettavan tiedoston rakenne näkyy alla, ohjelma kysyy tallennettavan tiedoston nimen.

```
Tehtävä;Lkm
L01T1;211
L01T2;206
L01T3;206
L01T4;206
L01T5;205
L02T1;200
L02T2;198
...
```

## 5. Esimerkkiajo

```
1) Lue tiedosto
2) Analysoi tiedot
3) Tulosta tulokset
0) Lopeta
Valintasi: 1
Anna tiedoston nimi: data15.csv
Yhteensä 15 palautusta 08.09.2020 - 11.02.2021 väliseltä ajalta.

1) Lue tiedosto
2) Analysoi tiedot
3) Tulosta tulokset
0) Lopeta
Valintasi: 2
Palautuksia oli yhteensä 15, 13 eri tehtävässä, keskimäärin 1 per tehtävä.

1) Lue tiedosto
2) Analysoi tiedot
3) Tulosta tulokset
0) Lopeta
Valintasi: 3
Tulosta tiedostoon (k/e): k
Anna tiedoston nimi: tulokset_lyhyt.csv

1) Lue tiedosto
2) Analysoi tiedot
3) Tulosta tulokset
0) Lopeta
Valintasi: 4
Tuntematon valinta, yritä uudestaan.

1) Lue tiedosto
2) Analysoi tiedot
3) Tulosta tulokset
0) Lopeta
Valintasi: 0
Kiitos ohjelman käytöstä.
```

## 6. Virheilmoitukset

Esimerkkiajossa ei näy kaikki ohjelman virheilmoitukset, joten ne on koottu alle. Käytä niitä tarpeen mukaan.

- "Tuntematon valinta, yritä uudestaan.\n"
- "Muistinvaraus epäonnistui"
- "Tiedoston lukeminen epäonnistui"
- "Tiedostoon kirjoittaminen epäonnistui"
- "Ei analysoitavaa, lue ensin palautustiedosto.\n"
- "Ei tulostettavaa, analysoi ensin palautustiedosto.\n"