

## L6 Tehtävät

- Dynaaminen muistihallinta, varaaminen, vapauttaminen linkitetyn listan yhteydessä
- Aliohjelmat, valikkopohjainen ohjelma, tietueet
- Tehtävät 1-3 antavat hyvän pohjan harjoitustyölle ja T4 on tavoitetason tehtävä

Lue oppaan luku 6, jossa on käsitelty nämä asiat.

L6T1: Linkitetyn listan perusoperaatiot .....	1
L6T2: Linkitetty lista aliohjelmilla .....	1
L6T3: Linkitetty lista aliohjelmilla valikkopohjaisessa ohjelmassa .....	2
L6T4: Valikkopohjainen ohjelma listan hallintaan .....	3

### L6T1: Linkitetyn listan perusoperaatiot

Tee yksinkertainen ohjelma, joka kysyy käyttäjältä lukuja ja lisää ne linkitettyyn listaan, kunnes käyttäjä lopettaa lisäämisen antamalla luvun 0. Tämän jälkeen ohjelma tulostaa listaan laitettut luvut, vapauttaa varatut muistialueet ja lopettaa ohjelman suorituksen. Määrittele ohjelmaan sopiva tietue ja toteuta tämä ohjelma ilman aliohjelmia. Ohjelmassa tulee olla itse määritelty tietue ja siitä tehty tietotyyppi; käytä listan tulostuksessa formaattia "%d ".

#### Esimerkkiäjo:

```
Anna listan luvut.
Anna positiivinen kokonaisluku, 0 lopettaa: 2
Anna positiivinen kokonaisluku, 0 lopettaa: 3
Anna positiivinen kokonaisluku, 0 lopettaa: 1
Anna positiivinen kokonaisluku, 0 lopettaa: 0
Listassa on seuraavat luvut: 2 3 1
Kiitos ohjelman käytöstä.
```

### L6T2: Linkitetty lista aliohjelmilla

Tee ohjelma, joka lukee tekstitiedostosta auton merkin ja vuosiluvun ja lisää nämä linkitettyyn listaan omina alkioinaan. Kun tiedosto on luettu, tulosta tiedot näytölle ja vapauta varattu muisti. Anna tiedoston nimi ohjelmalle komentoriviparametrina. Alla on esimerkki tiedostosta (tiedot.txt), josta tiedot luetaan. Huomaa, että eri tehtävissä sisältö voi olla erilainen. Luettava tiedosto voi myös sisältää tyhjiä rivejä.

```
Lada 1976
Ferrari 2005
Suzuki 1985
Volvo 1963
Toyota 1993
Honda 2011
```

Ohjelmassa tulee olla ohjelman kulun määrittävän pääohjelman lisäksi 3 aliohjelmaa. Yhdessä aliohjelmassa suoritetaan muistin varaaminen uudelle tietueelle ja arvojen asettaminen sekä tietueen lisääminen listaan viimeiseksi alkioiksi. Toinen aliohjelma huolehtii listan tietojen tulostamisesta ja kolmas muistin vapauttamisesta. Käsiteltäessä muistialkioita aliohjelmissa kannattaa lähettää osoite kiinnostuksen kohteena olevaan alkioon parametrina ja palauttaa sopivan alkion osoite paluuarvona.

Laita ohjelmaan seuraavat virheilmoitukset sopiviin kohtiin:

1. "Et antanut tiedoston nimeä."
2. "Tiedoston avaus epäonnistui."
3. "Muistinvaraus epäonnistui"
4. "Ajoneuvoja ei löytynyt.\n" tyhjän listan tapauksessa.

**Esimerkkiajo:**

```
Luetaan tiedot tiedostosta tiedot.txt.  
Tiedot luettu linkitettyyn listaan.  
1. ajoneuvo: Lada vuosimalli 1976.  
2. ajoneuvo: Ferrari vuosimalli 2005.  
3. ajoneuvo: Suzuki vuosimalli 1985.  
Muisti vapautettu.  
Kiitos ohjelman käytöstä.
```

**L6T3: Linkitetty lista aliohjelmilla valikkopohjaisessa ohjelmassa**

Toteuta tehtävä L5T4 linkitettyinä listana taulukon sijaan. Huomaa, että joudut muuttamaan ohjelman rakennetta jonkin verran ja lisäämään sopivia aliohjelmiä ohjelmaan L6T2 tehtävän mukaisesti. Toteuta ohjelma useiden aliohjelmien avulla ja pidä kunkin aliohjelman koko alle 20 rivissä ellei pidemmän ohjelman toteutukseen ole painavat perusteet. Muuttaessasi listan pituutta voit halutessasi ensin tyhjentää sen ja luoda sitten uudestaan. Alla on ohjelman virheilmoitukset ja edellisen viikon tehtävänanto.

Virheilmoitukset:

1. "Lista on tyhjä.\n"
2. "Muistin varaus epäonnistui"
3. "Listan pituus ei voi olla negatiivinen.\n"
4. "Tuntematon valinta, yritä uudestaan.\n"

**L5T4:**

*Tee valikkopohjainen ohjelma, joka pystyy tulostamaan kokonaislukutaulukossa olevat luvut ja muuttamaan käytössä olevan taulukon kokoa, kunnes käyttäjä haluaa lopettaa ohjelman suorituksen. Ohjelman alussa tulee luoda osoitin kokonaislukutaulukkoon ja taulukon kokoa ylläpitävä muuttuja, jonka oletusarvo on nolla. Kun käyttäjä muuttaa taulukon kokoa, varaa muistia dynaamisesti tarvittava määrä sekä täytää sen numeroilla 0-N (taulukon koko). Tulostettaessa ohjelma tulostaa kaikki taulukossa olevat arvot samalla riville välilyönnillä erotettuna. Muistin varaus ja tulostaminen tulee suorittaa omissa aliohjelmissaan. Muista varmistaa aina muistivarauksen onnistuminen, vapauta varattu muisti ohjelman lopuksi ja pyri käymään läpi kaikki vaihtoehdot käyttäjän syöttämille arvoille.*

**Esimerkkiajo:**

```
1) Tulosta listan alkiot  
2) Muuta listan pituutta  
0) Lopeta  
Anna valintasi: 2  
Anna listan uusi pituus: 4  
1) Tulosta listan alkiot  
2) Muuta listan pituutta  
0) Lopeta  
Anna valintasi: 1  
Listassa on seuraavat luvut:  
0 1 2 3  
1) Tulosta listan alkiot  
2) Muuta listan pituutta
```

```
0) Lopeta
Anna valintasi: 0
Kiitos ohjelman käytöstä.
```

## L6T4: Valikkopohjainen ohjelma listan hallintaan

Tee valikkopohjainen ohjelma listan hallintaan. Listassa tulee olla seuraavat toiminnallisuudet: listan luominen, alkion lisääminen listan loppuun, listan tyhjennys, listan tulostus ja ohjelman lopetus. Alla olevan esimerkkiajon valikossa on muutama lisätoiminnallisuus, jotka toteutetaan seuraavan viikon tehtävänä, tällä viikolla riittää siis alla olevassa esimerkkiohjelmassa testatut toiminnallisuudet. Tulevat toiminnallisuudet saattavat kuitenkin vaikuttaa ohjelman järkevään aliohjelmajakoon, joten ne kannattaa pitää mielessä ohjelman rakennetta toteuttaessa. Mutta koko ohjelman voi myös myöhemmin muuttaa laajennusten edellyttämään muotoon (refactoring), joten saat itse päättää toteutusstrategiasi. Tällä viikolla testattavat ominaisuudet perustuvat luennoilla ja esimerkkiohjelmissa käsiteltyihin asioihin, mutta tulevat ominaisuudet edellyttävät osoittimien hyvää hallintaa. Näin ollen tämä ohjelma on yksi tämän kurssin haastavimpia ja samalla opettavaisimpia tehtäviä.

Ohjelman rakenne: Ohjelma kannattaa aloittaa pääohjelman toteuttamisella ja lisää siihen tarvittavia aliohjelmaa sitä mukaa kun ohjelman kehitys etenee. Listan luomiseen voi tehdä yhden aliohjelman, mutta kannattaa myös harkita sitä, että tekee aliohjelman alkion lisäämiseen listaan siten, että samaa funktiota voi käyttää kaikissa kolmessa ensimmäisessä valikon toiminnallisuudessa. Toinen vakavan harkinnan paikka on kiinnostuksen kohteena olevan alkion löytäminen listasta, sillä oikein toteutettuna sitä voi käyttää hyväksi sekä alkiodien lisäämisen että poistamisen yhteydessä. Listan solmussa tulee olla tietona kokonaisluku, joka on juokseva numero 1 eteenpäin esimerkkiajon mukaisesti. Kasvata tätä solmun numeroa aina yhdellä, kun uusi alkio lisätään listaan eikä sitä nollata missään vaiheessa. Tulosta lista formaatilla "%d ".

Esimerkkiajossa olevat virheilmoitukset, kaikkia ei välttämättä tarvita/testata:

1. "Muistin varaus epäonnistui"
2. "Indeksivirhe, keskeytetään lisäys.\n"
3. "Järjestys ei voi olla negatiivinen.\n"
4. "Liian iso järjestysnumero, ei alkiota.\n"
5. "Järjestysnumeron pitää olla suurempi kuin nolla.\n"
6. "Järjestys ei voi olla alle 1.\n"
7. "Tunteen valinta, yritä uudestaan.\n"

### Esimerkkiajo:

Tämä ohjelma hallitsee listaa ja sen alkioita.

```
1) Luo lista
2) Lisää alkio listan loppuun
3) Lisää alkio listan keskelle
4) Tyhjennä lista
5) Poista alkio listasta
6) Tulosta lista
0) Lopeta
Anna valintasi: 1
Anna listan pituus: 5
1 2 3 4 5
1) Luo lista
2) Lisää alkio listan loppuun
3) Lisää alkio listan keskelle
```

```
4) Tyhjennä lista
5) Poista alkio listasta
6) Tulosta lista
0) Lopeta
Anna valintasi: 4
1) Luo lista
2) Lisää alkio listan loppuun
3) Lisää alkio listan keskelle
4) Tyhjennä lista
5) Poista alkio listasta
6) Tulosta lista
0) Lopeta
Anna valintasi: 6
1) Luo lista
2) Lisää alkio listan loppuun
3) Lisää alkio listan keskelle
4) Tyhjennä lista
5) Poista alkio listasta
6) Tulosta lista
0) Lopeta
Anna valintasi: 0
Kiitos ohjelman käytöstä.
```