

## L3 Tehtävät

- Tiedostonkäsittely: Lukeminen ja kirjoittaminen, teksti- ja binaaritiedostot
- Pääohjelma, aliohjelma, funktio
- Arvo- ja muuttujaparametrit, paluuarvot
- Kirjastojen käyttö

Lue oppaan luku 3, jossa on käsitelty nämä asiat.

L3T1: Funktio parametreilla ja paluuarvolla, potenssiin korotus.....	1
L3T2: Valikkopohjainen ohjelma tiedostonkäsittelyyn aliohjelmilla.....	1
L3T3: Funktiot ja tietotyypit.....	2
L3T4: Merkkijonon kopiointi itse tehdyllä funktiolla .....	3
L3T5: Satunnaislukujen kirjoittaminen binaaritiedostoon.....	4

### L3T1: Funktio parametreilla ja paluuarvolla, potenssiin korotus

Tee ohjelma, joka laskee luvun potenssin aliohjelmassa. Kysy luku ja potenssi käyttäjältä pääohjelmassa, välitä ne aliohjelmalle parametreina, suorita laskenta toistorakenteen avulla ja palauta laskun tulos aliohjelmasta pääohjelmaan paluuarvona. Tulosta tulos pääohjelmassa alla olevan esimerkin mukaisesti.

#### Esimerkkiajo:

```
Ohjelma laskee luvun potenssin.  
Anna kokonaisluku, jonka haluat korottaa potenssiin: 5  
Anna potenssi, jonka haluat laskea: 4  
5 ^ 4 = 625.  
Kiitos ohjelman käytöstä.
```

### L3T2: Valikkopohjainen ohjelma tiedostonkäsittelyyn aliohjelmilla

Tee ohjelma, jossa on pääohjelman lisäksi kolme aliohjelmaa – yksi valikkoa varten, toinen tiedoston kirjoittamiseen ja kolmas lukemiseen. Tiedostoa kirjoittava aliohjelma kysyy käyttäjältä nimiä ja lisää ne tiedostoon siellä aiemmin olevien nimien jatkoksi. Tiedostoa lukeva aliohjelma avaa tiedoston, lukee nimet ja tulostaa ne näytölle riveittäin. Tee valikko ohjelman pyörittämiseen ja käytä valikon toteuttamiseen if, else-if, else -rakennetta. Ohjelma kysyy alussa käsiteltävän tiedoston nimen ja välitä sen parametrina aliohjelmiin. Käyttäjältä kysyttävien nimien maksimipituus on 48 merkkiä, tee sitä varten ohjelmaan oma vakio. Muista laittaa tiedostonkäsittelyihin virheenkäsittely.

#### Esimerkki nimitiedostosta nimet.txt:

```
Petteri  
Jarno  
Mika
```

**Esimerkkiajo:**

Tämä ohjelma lisää nimiä tiedostoon ja lukee ne.

Anna käsiteltävän tiedoston nimi: nimet.txt

Valitse alla olevista valinnoista

1) Lisää uusi nimi

2) Tulosta nimet

0) Lopeta

Anna valintasi: 1

Anna lisättävä nimi: Brian

Nimi lisätty tiedostoon.

Valitse alla olevista valinnoista

1) Lisää uusi nimi

2) Tulosta nimet

0) Lopeta

Anna valintasi: 2

Tiedostossa olevat nimet:

Petteri

Jarno

Mika

Brian

Tiedosto luettu ja tulostettu.

Valitse alla olevista valinnoista

1) Lisää uusi nimi

2) Tulosta nimet

0) Lopeta

Anna valintasi: 0

Kiitos ohjelman käytöstä.

**Esimerkkiajo 2:**

Tämä ohjelma lisää nimiä tiedostoon ja lukee ne.

Anna käsiteltävän tiedoston nimi: eiote.txt

Valitse alla olevista valinnoista

1) Lisää uusi nimi

2) Tulosta nimet

0) Lopeta

Anna valintasi: 2

Tiedoston avaaminen epäonnistui, lopetetaan: No such file or directory

**L3T3: Funktiot ja tietotyypit**

Tee ohjelma, joka kysyy 2 kokonaislukua, selvittää itse tehdyllä aliohjelmalla kumpi luvuista on suurempi ja pienempi sekä tulostaa sen jälkeen nämä näytölle. Toista sama kahdelle desimaaliluvulle ja mieti montako aliohjelmaa sinun kannattaa tehdä – riittääkö 1 tai 2 vai kannattaako tehdä joku muu määrä aliohjelmia? Katso esimerkkiajosta yksityiskohdat ja huomaa, että desimaalien tulostuksessa käytetään aina %5.2f formaattia.

Lisää pääohjelman loppuun merkkijonojen vertailu siten, että pääohjelmassa vertaillaan ensin kokonaislukuja, sitten desimaalilukuja ja kolmantena merkkijonoja. Kysy kaksi nimeä, max 20 merkkiä, ja tulosta näytölle miten nämä nimet suhtautuvat toisiinsa merkkijonovertailufunktion strcmp-mukaan eli kumpi on suurempi ja pienempi vai ovatko ne yhtä suuria. Mikäli strcmp-funktio ei ole tuttu, katso sen toiminta man-sivuilta.

**Esimerkkiajo 1:**

Anna kaksi kokonaislukua:  
Luku 1: 5  
Luku 2: 9  
Luku 9 on suurempi ja 5 pienempi.  
Anna kaksi desimaalilukua:  
Luku 1: 8.2  
Luku 2: 24.2315  
Luku 24.23 on suurempi ja 8.20 pienempi.  
Anna kaksi nimeä:  
Nimi 1: Ville  
Nimi 2: Kalle  
Merkkijonona 'Ville' on suurempi ja 'Kalle' pienempi.  
Kiitos ohjelman käytöstä.

**Esimerkkiajo 2:**

Anna kaksi kokonaislukua:  
Luku 1: 3  
Luku 2: 3  
Luvut 3 ja 3 ovat yhtä suuria.  
Anna kaksi desimaalilukua:  
Luku 1: 28.1  
Luku 2: 28.1  
Luvut 28.10 ja 28.10 ovat yhtä suuria.  
Anna kaksi nimeä:  
Nimi 1: Erkki  
Nimi 2: Erkki  
Merkkijonona 'Erkki' ja 'Erkki' ovat yhtä suuria.  
Kiitos ohjelman käytöstä.

**L3T4: Merkkijonon kopiointi itse tehdyllä funktiolla**

Tee ohjelma, joka kysyy käyttäjältä merkkijonon, kopio sen toiseen merkkijonomuuttujaan ja lopuksi tulostaa molemmat merkkijonot päällekkäin näytölle. Käytä merkkijonon lukemiseen `fgets()`-funktia, jolloin saat koko rivin välilyöntien kanssa ja muista huomioida rivinvaihtomerkki. Suorita merkkijonon kopiointi muistipaikkojen välillä **käyttämättä `strcpy`-funktia ja `string.h`-kirjastoa**. Tämä onnistuu, kun teet kaksi omaa aliohjelmaa, joista

1. yksi laskee parametrina tulevan merkkijonon pituuden (ts. vastaa `strlen`-funktia, ks. man)
2. toinen saa kaksi merkkitaulukkoa parametreinä ja se kopioi toisena parametrina olevan merkkitaulukon sisällön ensimmäisenä parametrina olevaan merkkitaulukkoon yksi merkki kerrallaan (ts. vastaa `strcpy`-funktia, ks. man).

Mallia voit katsoa esim. oppaan luvun 2 esimerkeistä 2.11 ja 2.12, joista ensimmäisessä käytetään indeksejä ja toisessa osoitinta merkkijonon kanssa. Muista kopioida uuteen merkkijonoon myös loppu-merkki. Toteuta kaikki syöttö ja tulostus pääohjelmassa, pituuden laskeva funktio palauttaa pituuden ja kopioinnin tekevä funktio palauttaa osoittimen kopioituun (ts. uuteen) merkkijonoon. Molempien merkkitaulukoiden koko on 30 merkkiä ja määrittele kokoa varten oma vakio.

**Esimerkkiajo 1:**

```
Anna kopioitava merkkijono: Erkki
Merkkijono 1 on 'Erkki'.
Merkkijono 2 on 'Erkki'.
Kiitos ohjelman käytöstä.
```

**Esimerkkiajo 2:**

```
Anna kopioitava merkkijono: Tämä on merkkijono.
Merkkijono 1 on 'Tämä on merkkijono.'.
Merkkijono 2 on 'Tämä on merkkijono.'.
Kiitos ohjelman käytöstä.
```

**L3T5: Satunnaislukujen kirjoittaminen binaaritiedostoon**

Tee ohjelma, joka kirjoittaa ja lukee tiedoston binäärimuodossa. Ohjelman tulee ensin arpoa 20 kokonaislukua, tallentaa ne binäärimuodossa tiedostoon ja lukea sen jälkeen tallennettu tiedosto sekä tulostaa siellä olevat luvut näytölle.

Kysy käyttäjältä pääohjelman alussa ensin käsiteltävän binaaritiedoston nimen ja satunnaislukugeneraattorin lähtöarvon sekä välittää ne kirjoittavaan aliohjelmaan. Tämän tulee ensin alustaa satunnaislukugeneraattori `srand()` -funktioilla ja käyttäjän antamalla arvolla, jonka jälkeen se avaa tiedoston, arpoo luvut ja kirjoittaa ne tiedostoon sekä sulkee tiedoston. Satunnaisluvut arvotaan `rand()` -funktioilla väliltä 0-999. Mikäli tiedoston kirjoitus onnistuu, kutsuu pääohjelma seuraavaksi aliohjelmaa, joka avaa kirjoitetun tiedoston sekä lukee ja tulostaa siellä olevat luvut näytölle.

Tarkempia tietoja `srand` ja `rand` -funktioista löytyy manuaalista. Mikäli tiedoston avaus epäonnistuu, tulee siinä ole ohjelmointioppaan mukainen virheenkäsittely ja ohjelman lopetus, ks. alla oleva esimerkki. Sekä tiedoston kirjoitus- että lukualiohjelmien tulee palauttaa tieto pääohjelmaan tiedoston avaamisen onnistumisesta ja pääohjelmassa tulostetaan käyttäjälle tieto asiasta tyyliin "Tiedoston kirjoitus/lukeminen onnistui."

**Esimerkkiajo 1:**

```
Anna käsiteltävän tiedoston nimi: luvut.bin
Anna satunnaisluvuille lähtöarvo: 1
Tiedoston kirjoitus onnistui.
Tiedostossa on seuraavat luvut:
0 966 695 955 588 850 754 700 628 452 59 679 999 421 176 878 20 165 683 84
Tiedoston lukeminen onnistui.
Kiitos ohjelman käytöstä.
```

**Esimerkkiajo 2:**

```
Anna käsiteltävän tiedoston nimi: /var/eiole.txt
Anna satunnaisluvuille lähtöarvo: 1
Tiedoston avaaminen epäonnistui, lopetetaan: Permission denied
```