

## L02 Tehtävät

- Perustietotyypit: int, float, char, char []
- Tietojen kysyminen ja tulostaminen
- Toistorakenteet: for, while
- Valintarakenteet: if-else if - else, switch - case
- Hypyty: return, break

Lue oppaan luku 2, jossa on käsitelty nämä asiat.

L2T1: Valintarakenne, valikkopohjainen ohjelma .....	1
L2T2: Toistorakenne for, virheenkäsittely .....	1
L2T3: Toistorakenne while, virheenkäsittely .....	2
L2T4: Valikkopohjainen ohjelma switch, vakiot .....	2
L2T5: Valikkopohjainen ohjelma, merkkitaulukon käsittely, vakio, virheenkäsittely .....	3

### L2T1: Valintarakenne, valikkopohjainen ohjelma

Tee ohjelma, joka kysyy käyttäjältä kaksi lukua ja tekee niille laskuoperaation yksinkertaisen valikon perusteella. Vaihtoehdot ovat summaa luvut yhteen ja kerro luvut keskenään. Suorita valinnan perusteella edellä mainitut operaatiot käyttäen hyväksi if - else if - else -rakennetta. Tulosta tehdyn operaation kuvaus ja tulos alla olevan esimerkin mukaisesti. Palauta ohjelman lähdekoodi tiedostona ”L2T1.c” CodeGradeen

#### Esimerkkiajo 1

Anna kaksi kokonaislukua: 5 8

Valikko

1) Summaa luvut yhteen.

2) Kerro luvut keskenään.

Valitse: 1

Luvut laskettiin yhteen. Tulos = 13.

#### Esimerkkiajo 2

Anna kaksi kokonaislukua: 3 8

Valikko

1) Summaa luvut yhteen.

2) Kerro luvut keskenään.

Valitse: 6

Tunteamaton valinta.

### L2T2: Toistorakenne for, virheenkäsittely

Tee ohjelma, joka kysyy käyttäjältä luvun väliltä 10 – 200, laskee yhteen luvut nolasta kysyttyyn lukuun ja tulostaa vastauksen alla olevan esimerkin mukaisesti. Muista tarkistaa, että luku on annettulla välillä ja huomauta käyttäjälle, mikäli luku ei ole oikealla välillä. Käytä summan laskemiseen for rakennetta. Palauta ohjelman lähdekoodi tiedostona ”L2T2.c” CodeGradeen

**Esimerkkiajo 1**

Anna jokin luku väliltä 10 - 200: 65  
Lukujen 0 - 65 summa on 2145.

**Esimerkkiajo 2**

Anna jokin luku väliltä 10 - 200: 265  
Antamasi luku ei ole annetulla välillä.

**L2T3: Toistorakenne while, virheen käsittely**

Tee ohjelma, joka kysyy käyttäjältä kokonaisluvun väliltä 1 – 10 ja tulosta kaikki luvut väliltä 1 ja annettu luku alla olevan esimerkin mukaisesti. Tarkista, että annettu luku on oikealla välillä ja käytä silmukassa `while` rakennetta. Palauta ohjelman lähdekoodi tiedostona ”L2T3.c” CodeGradeen

**Esimerkkiajo 1**

Anna jokin kokonaisluku väliltä 1-10: 6  
1 kierros.  
2 kierros.  
3 kierros.  
4 kierros.  
5 kierros.  
6 kierros.

**Esimerkkiajo 2**

Anna jokin kokonaisluku väliltä 1-10: 25  
Antamasi luku ei ole välillä 1-10.

**L2T4: Valikkopohjainen ohjelma switch, vakiot**

Tee ohjelma, joka kysyy käyttäjältä liukuluvun. Tee sitten yksinkertainen valikko, jossa on kaksi vaihtoehtoa: (1) "Kerro piin arvo luvulla xx" ja (2) "Jaa piin arvo luvulla xx" siten, että luku xx on käyttäjän antama luku. Käytä valintarakenteena `switch-case` rakennetta. Määritä piin likiarvo vakioksi esikäsitteilyjä avulla ja käytä sitä laskutoimituksissa. Käytä piin arvossa ja liukuluvuissa kolmen desimaalin tarkkuutta (esim. 3.142). Jätä tulosteissa liukulukujen kokonaisosan leveys määrittämättä eli käytä formaattia `% .3f`. Palauta ohjelman lähdekoodi tiedostona ”L2T4.c” CodeGradeen

**Esimerkkiajo 1**

Anna liukuluku: 1.7  
  
Valikko  
1) Kerro piin arvo luvulla 1.700.  
2) Jaa piin arvo luvulla 1.700.  
Valitse: 1  
  
Pii \* 1.700 = 5.341.

**Esimerkkiajo 2**

Anna liukuluku: 2.4  
  
Valikko  
1) Kerro piin arvo luvulla 2.400.

```
2) Jaa piin arvo luvulla 2.400.  
Valitse: 5
```

```
Tuntematon valinta.
```

## L2T5: Valikkopohjainen ohjelma, merkkitaulukon käsittely, vakio, virheenkäsittely

C-kielessä merkkijonojen käsittely perustuu varattuun muistialueeseen eli merkkitaulukkoon ja loppumerkkiin. Tee ohjelma, jossa merkkijonon käsittely on toteutettu toisella tavalla siten, että varaat merkkitaulukon ohjelmassasi ja ylläpidät muuttujassa merkkijonon pituutta eikä merkkijonossa siis olekaan loppumerkkiä. Ohjelman tulee pyöriä niin kauan, kunnes käyttäjä haluaa lopettaa sen. Ohjelmassa tulee olla seuraava valikko:

- 1) Lisää uusi merkki
- 2) Tyhjennä merkkijono
- 3) Tulosta merkkijono
- 0) Lopeta

Varaa ohjelman alussa merkkitaulukko, jonka koko on 200 merkkiä käyttäen vakiota taulukon koon määrittämiseen. Toteuta yllä olevat toiminnot ohjelmassasi `switch-case` rakennetta käyttäen.

Merkin lisääminen tarkoittaa merkin tallettamista taulukon seuraavan vapaan merkin kohdalle; taulukon tyhjentäminen tarkoittaa taulukossa olevien merkkien ylikirjoittamista välilyönnillä ja merkkijonon pituuden asettamista nolllaksi; tulostettaessa kaikki taulukossa olevat merkit tulostetaan yksi kerrallaan merkkijonon pituuden mukaisesti.

Toteuta ohjelmassa virheentarkistus käyttäen `if else` rakennetta. Tarkista valikon mahdollinen väärä syöte. Mikäli merkkijonoon ei mahdu enempää merkkejä, ilmoita siitä käyttäjälle. Jos merkkijonossa ei ole merkkejä, kun sitä tulostetaan tai tyhjennetään, ilmoita siitä käyttäjälle. Käytä ohjelmassa seuraavia virheilmoituksia:

- Tuntematon valinta.
- Merkkijonoon ei mahdu enempää merkkejä.
- Merkkijono on tyhjä.

Huomaa, että ohjelmassa ei käytetä `string.h` -otsikkotiedostoa vaan `stdio.h` on ainoa tarvittava otsikkotiedosto. Palauta ohjelman lähdekoodi tiedostona ”L2T5.c” CodeGradeen

### Esimerkkiajo

```
Valikko
```

- 1) Lisää uusi merkki
- 2) Tyhjennä merkkijono
- 3) Tulosta merkkijono
- 0) Lopeta

```
Valintasi: 1
```

```
Anna jokin merkki: A
```

```
Valikko
```

- 1) Lisää uusi merkki
- 2) Tyhjennä merkkijono
- 3) Tulosta merkkijono

0) Lopeta

Valintasi: 3

Merkkijono: A

Valikko

1) Lisää uusi merkki

2) Tyhjennä merkkijono

3) Tulosta merkkijono

0) Lopeta

Valintasi: 2

Merkkijono on tyhjennetty.

Valikko

1) Lisää uusi merkki

2) Tyhjennä merkkijono

3) Tulosta merkkijono

0) Lopeta

Valintasi: 5

Tuntematon valinta.

Valikko

1) Lisää uusi merkki

2) Tyhjennä merkkijono

3) Tulosta merkkijono

0) Lopeta

Valintasi: 0

Ohjelman lopetus.