

ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

4. huhtikuuta 2025

Yleistä

Tentti oli pistekeskiarvon 12,9 (keskihajonta 6,7) perusteella keskitasoa, joskin hajonta oli melko suurta. Demohyvitysten kanssa keskiarvo oli 17,8. Huomaa, että demopisteet on laskettu tentin päälle, ja arvosana lasketaan vasta sen jälkeen. Kysymykset löytyvät liitteestä A. Opiskelijan omat tehtävät ovat nähtävissä TIMissä alkuperäisellä tenttisivulla. Uusintojen ajankohdat löydät Sisusta.

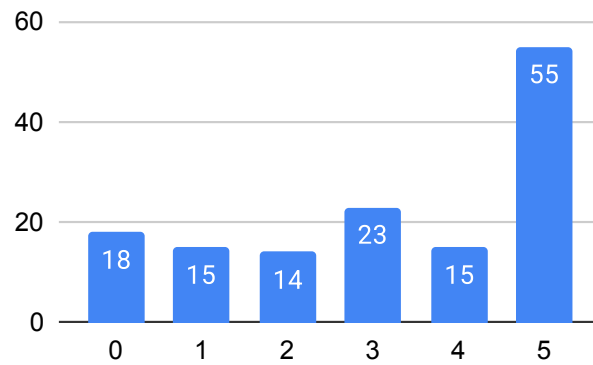
Tehtävä	Teki	Keskiarvo	Automaattinen arvostelu / Tarkastaja
T1	140	3,4	Automaattinen
T2	140	3,5	Automaattinen
T3	140	3,8	Automaattinen
T4	140	2,3	Automaattinen
Yht		12,9	(ennen demopisteitä)



JYVÄSKYLÄN YLIOPISTO

Arvosteluasteikko ja arvosanajakauma

Arvolause	Pistemäärä (inklusiivinen alaraja)
1	10
2	13
3	15
4	18
5	21



Tentin arvosanajakauma.

Tehtävä 1 (6 p.)

Oikeat vastaukset on lihavoitu.

- Kysymys 1.1: Mikä seuraavista väitteistä pitää paikkansa? C#-ohjelmassa...
 - tulee olla Main:n lisäksi vähintään yksi aliohjelma.
 - Main:ssa tulee olla vähintään yksi return-lause.
 - **Main:n sisällä oleva koodi suoritetaan aina ensimmäisenä.**
 - Main:n sisällä oleva koodi suoritetaan vain, jos Main-funktiota kutsutaan toisesta aliohjelmasta.
- Kysymys 1.2: Mikä seuraavista väitteistä pitää paikkansa?
 - Ohjelman voi ajaa ilman kääntämistä
 - Ohjelman voi ajaa, vaikka se **ei** kääntyisi
 - Ohjelman voi ajaa, vaikka se sisältäisi syntaksivirheitä
 - **Ohjelman voi ajaa, vaikka se sisältäisi loogisia virheitä**
- Kysymys 1.3: Oletetaan, että meillä on käytettävissä seuraavat tyypit: int, double, string, bool ja char. Puhuttaessa tyypeistä tarkoitetaan siis tässä kysymyksessä näitä viittä tyyppiä. Mikä seuraavista väitteistä pitää paikkansa?
 - Ohjelmoija voi muuttaa muuttujan tyypin ohjelman suorituksen aikana
 - Muuttujan tyyppi riippuu siitä, minkä tyyppinen arvo siihen on sijoitettu
 - **Muuttujaan sijoitettu arvo voidaan muuttaa toisen tyyppiseksi ohjelman suorituksen aikana**
 - Muuttujaan sijoitettu arvo voidaan muuttaa toisen tyyppiseksi vain, jos muuttujan tyyppi on string
- Kysymys 1.4: Mikä seuraavista väitteistä pitää paikkansa koskien alla olevaa koodia? (Tehtävästä puuttui virheellisesti luokka, joten hyväksyttiin kaksi eri vastausta.)

```
public static void Main()
{
    // Tämä koodi tulostaa "Hello world"
    /*
    Console.WriteLine("Hello world");
    */
    /// Yläpuolinen koodi tulostaa "Hello world"
}
```

- **Ohjelman voi kääntää ja ajaa.**
- Ohjelma tulostaa "Hello world"

- Ohjelma sisältää dokumentaatiokommentin
- **Ohjelma *ei* sisällä mitään käännettävää koodia**
- Kysymys 1.5: Mikä seuraavista väitteistä pitää paikkansa koskien alla olevaa koodia?

```

01 public static void Main()
02 {
03     int a = 3;
04     Muuta(a);
05     a++;
06     int b = 4;
07     b = a++;
08     Muuta(b);
09 }
10
11 public static void Muuta(int a)
12 {
13     a++;
14     return;
15 }

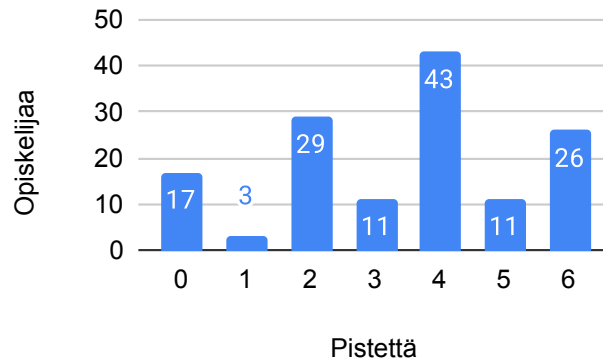
```

- Tullessa riville 5, muuttujan a arvo on 4
- Kun ohjelma päättyy, muuttujien a ja b arvot ovat samat
- **Aliohjelma Muuta muuttaa parametrina saamaansa arvoa**
- Ohjelma toimisi samalla tavalla, vaikka rivi 4 korvattaisiin rivillä a++;
- Kysymys 1.6: Mitä break-lauseen jälkeen tapahtuu?
 - **Poistutaan silmukasta**
 - Keskeytetään ohjelman suoritus
 - Palautetaan aliohjelman paluuarvo
 - Jatketaan suoraan silmukan seuraavalle kierrokselle

Tehtävä 2 (6 p.)

Oikeat vastaukset on lihavoitu.

- Kysymys 2.1: Mikä seuraavista väitteistä pitää paikkansa? (Kolmannen kohdan “voi”-sanon monitulkintaisuuden vuoksi tässä hyväksyttiin sekä ensimmäinen että kolmas vastausvaihtoehto.)
 - **Funktion dokumentaatiokommenteissa tulee kertoa parametrien nimet**
 - Dokumentaatiokommentit on tarkoitettu vain ohjelmoijalle itselleen



Kuva 1: Tehtävän 1 pistejakauma.

- **Funktion dokumentaatiokomentit voi kirjoittaa vasta kun funktio on valmis**
- Dokumentaatiokomentit kirjoitetaan funktion rajaavien aaltosulkujen väliin
- Kysymys 2.2: Oletetaan, että meillä on olemassa vain muuttujat jonot ja etsittava, jotka on alustettu seuraavasti:

```
List<string> jonot =
    new List<string>() { "kissa", "koira", "linnunpelätin" };
string etsittava = "koira";
```

Oletetaan lisäksi, että meillä on funktio, jonka esittelyrivi on:

```
public static List<string> Suodata(List<string> jonot, string hakusana)
```

Mikä seuraavista on syntaktisesti oikea tapa kutsua funktiota Suodata?

- Suodata("kissa", "koira")
- jonot = Suodata(jonot, hakusana)
- **Suodata(jonot, "etsittava")**
- Suodata(jonot, etsittava) = uudetJonot
- Kysymys 2.3:

Mikä seuraavista on totta, jos haluamme, että ohjelmassa ei ole käänkösvirheitä?

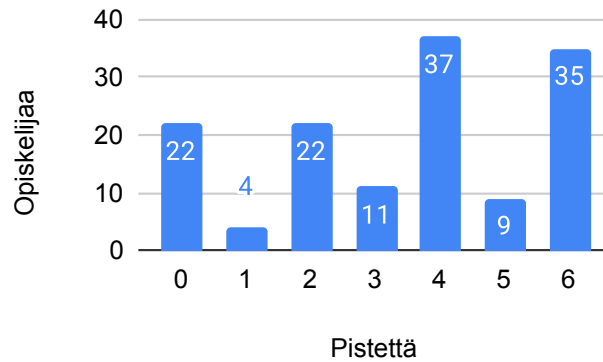
 - Ohjelmassa pitää olla rivinvaihtoja.
 - Ohjelmassa pitää olla vähintään yksi using-lause.
 - {-merkin jälkeen pitää tulla aina }-merkki ennen seuraavaa {-merkkiä.
 - **Avainsanojen välissä voi olla mielivaltaisen määrä välilyöntejä ja rivinvaihtoja.**

- Kysymys 2.4:
Oletetaan, että meillä on kaksi sisäkkäistä for-silmukkaa. Mikä seuraavista pitää paikkansa?
 - Sisempi silmukka suoritetaan aina vähintään yhden kerran.
 - Kummassakin silmukassa on käytettävä saman nimisiä muuttujia.
 - **Ulommassa silmukassa voi olla sisemmän silmukan jälkeen suoritettavia lauseita.**
 - Sisemmässä silmukassa olevat lauseet suoritetaan ennen ulomman silmukan lauseita.
- Kysymys 2.5: Alla on Funktio-niminen aliohjelma. Kysymykset 2.5 ja 2.6 liittyvät tähän aliohjelmaan.

```
public static int Funktio(int[] luvut)
{
    int yhteensa = 0;
    bool laskevaJono = false;
    for (int i = 1; i <= luvut.Length; i++)
    {
        if (i < luvut.Length && luvut[i] < luvut[i-1])
        {
            laskevaJono = true;
        }
        else if (laskevaJono)
        {
            yhteensa++;
            laskevaJono = false;
        }
    }
    return yhteensa;
}
```

Mikä seuraavista kuvaa parhaiten aliohjelman toimintaa? Funktio palauttaa...

- **laskevien osajonojen lukumäärän.**
- negatiivisten lukujen lukumäärän.
- laskevien osajonojen yhteispituuden.
- tiedon, sisältääkö taulukko laskevia osajonoja vai ei.
- Kysymys 2.6: Mikä seuraavista väitteistä on lähimpänä totuutta? Aliohjelma palauttaa arvon 0...
 - vain silloin, kun taulukko on tyhjä.
 - **silloin, kun taulukko on tyhjä, tai taulukko sisältää vain samoja lukuja.**
 - silloin, kun taulukko sisältää vain laskevan osajonon, esimerkiksi [3, 2, 1].
 - silloin, kun taulukko on tyhjä, tai taulukko sisältää vain negatiivisia lukuja.



Tehtävän 2 pistejakauma.

Tehtävä 3 (6 p.)

Arvostelu pohjautui automaattiseen arvioon Liitteessä B Taulukossa B.1 esitettyjen arvostelukohdeiden/testitapausten kautta. Automaattisilla arvostelukohdeilla on tärkeysjärjestys: Mikäli ratkaisu ei toteuta ylempänä olevaa kohdetta, ei sitä alempia arvostelukohdeita oteta huomioon. Pisteet pyöristettiin alaspäin lähimpään kokonaislukuun.

Malliratkaisu

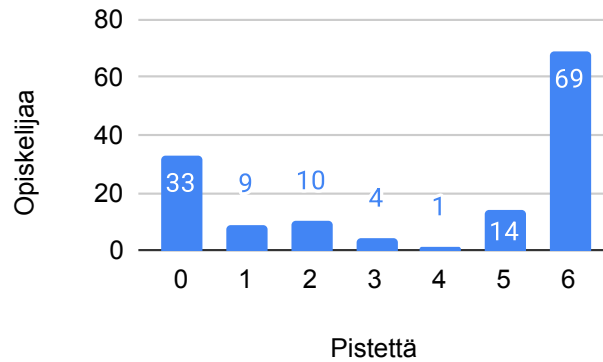
```
public static int[] VaalinEkaKierros(int[] aanet)
{
    int summa = 0;
    foreach (int a in aanet)
    {
        summa += a;
    }

    int suurin = aanet[aanet.Length - 1];
    if (aanet.Length == 1 || suurin > summa / 2)
    {
        return new int[] { suurin };
    }
    int toiseksiSuurin = aanet[aanet.Length - 2];
    return new int[] { toiseksiSuurin, suurin };
}
```

Hyväksyttiin myös ratkaisu, jossa toiseksi suurin ja suurin alkio olivat palautettavassa taulukossa toisin päin.

Tehtävä 4 (6 p.)

Arvostelu pohjautui automaattiseen arvioon Liitteessä B Taulukossa B.2 esitettyjen arvostelukohdeiden/testitapausten kautta. Automaattisilla arvostelukohdeilla on tärkeysjärjestys: Mikäli ratkaisu ei toteuta ylempänä olevaa kohdetta, ei sitä alempia ar-

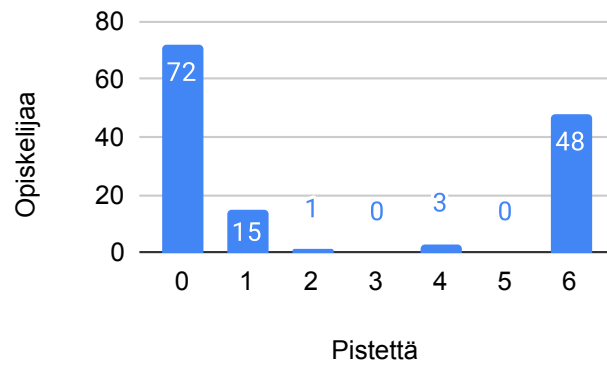


Tehtävän 3 pistejakauma.

vostelukohteita oteta huomioon. Pisteet pyöristettiin alaspäin lähimpään kokonaislukuun.

Malliratkaisu

```
public static int[] EriLuvut(int[] luvut1, int[] luvut2)
{
    List<int> eriLuvut = new List<int>();
    foreach (int luku in luvut1)
    {
        if (!luvut2.Contains(luku) && !eriLuvut.Contains(luku))
        {
            eriLuvut.Add(luku);
        }
    }
    foreach (int luku in luvut2)
    {
        if (!luvut1.Contains(luku) && !eriLuvut.Contains(luku))
        {
            eriLuvut.Add(luku);
        }
    }
    eriLuvut.Sort();
    return eriLuvut.ToArray();
}
```

Tehtävän 4 pistejakauma.

A Kysymykset

Ohjelmointi 1 (ITKP102), tentti pe 4.4.2025 klo 12:00-16:00

Tästä linkistä pääset takaisin tentin johdantosivulle. Voit palata tälle sivulle tenttiajan puitteissa niin monta kertaa kuin haluat.

Kaikki tehtävät koskevat C#-kieltä.

Ohjelmointitehtävissä on noudatettava tällä kursilla esitettyjä C#:n koodauskäytänteitä (muun muassa nimeäminen, sisennykset, dokumentaatiot, jne.)

Huomautus: Argumentti-sanalla viitataan aliohjelmakutsussa aliohjelmalle annettavaan arvoon. Parametri-sanalla viitataan aliohjelman määrittelyrivillä määriteltävään muuttujaan.

Plugin jsrunner error:

Plugin does not support printing yet. Please refer to TIM help pages if you want to learn how you can manually define what to print here.

Tehtävä 1 (6 p.)

Alla on monivalintakysymyksiä. Kuhunkin kysymykseen on täsmälleen yksi oikea vastaus. Arviointi:

- Oikea vastaus, +1 p.
- Väärä vastaus, -1 p.
- Ei vastattu, -1 p.
- Valittu "En halua vastata", 0 p.

Tehtävän kokonaispistemäärä on mahdollisista miinuspisteistä huolimatta vähintään 0.

Kysymys 1.1

Mikä seuraavista väitteistä pitää paikkansa? C#-ohjelmassa...

- ☐ tulee olla **Main**:n lisäksi vähintään yksi aliohjelma.
- ☐ **Main**:ssa tulee olla vähintään yksi **return**-lause.
- ☐ **Main**:n sisällä oleva koodi suoritetaan aina ensimmäisenä.
- ☐ **Main**:n sisällä oleva koodi suoritetaan vain, jos **Main**-funktiota kutsutaan toisesta aliohjelmasta.
- ☐ En halua vastata

Kysymys 1.2

Mikä seuraavista väitteistä pitää paikkansa?

- ☐ Ohjelman voi ajaa ilman kääntämistä
- ☐ Ohjelman voi ajaa, vaikka se *ei* kääntyisi
- ☐ Ohjelman voi ajaa, vaikka se sisältäisi syntaksivirheitä
- ☐ Ohjelman voi ajaa, vaikka se sisältäisi loogisia virheitä
- ☐ En halua vastata

Kysymys 1.3

Oletetaan, että meillä on käytettävissä seuraavat tyypit: `int`, `double`, `string`, `bool` ja `char`. Puhuttaessa tyypeistä tarkoitetaan siis tässä kysymyksessä näitä viittä tyyppiä.

Mikä seuraavista väitteistä pitää paikkansa?

- ☐ Ohjelmoija voi muuttaa muuttujan tyypin ohjelman suorituksen aikana
- ☐ Muuttujan tyyppi riippuu siitä, minkä tyyppinen arvo siihen on sijoitettu
- ☐ Muuttujaan sijoitettu arvo voidaan muuttaa toisen tyyppiseksi ohjelman suorituksen aikana
- ☐ Muuttujaan sijoitettu arvo voidaan muuttaa toisen tyyppiseksi vain, jos muuttujan tyyppi on `string`
- ☐ En halua vastata

Kysymys 1.4

Mikä seuraavista väitteistä pitää paikkansa koskien alla olevaa koodia?

```
public static void Main()
{
    // Tämä koodi tulostaa "Hello world"
    /*
    Console.WriteLine("Hello world");
    */
    /// Yläpuolinen koodi tulostaa "Hello world"
}
```

- ☐ Ohjelman voi kääntää ja ajaa.
- ☐ Ohjelma tulostaa "Hello world"
- ☐ Ohjelma sisältää dokumentaatiokommentin
- ☐ Ohjelma *ei* sisällä mitään käännettävää koodia
- ☐ En halua vastata

Plugin jsrunner error:

Plugin does not support printing yet. Please refer to TIM help pages if you want to learn how you can manually define what to print here.

Kysymys 1.5

Mikä seuraavista väitteistä pitää paikkansa koskien alla olevaa koodia?

```
01 public static void Main()
02 {
03     int a = 3;
04     Muuta(a);
05     a++;
06     int b = 4;
07     b = a++;
08     Muuta(b);
09 }
10
11 public static void Muuta(int a)
12 {
13     a++;
14     return;
15 }
```

- ☐ Tultaessa riville 5, muuttujan **a** arvo on 4
- ☐ Kun ohjelma päättyy, muuttujien **a** ja **b** arvot ovat samat
- ☐ Aliohjelma **Muuta** muuttaa parametrina saamaansa arvoa
- ☐ Ohjelma toimisi samalla tavalla, vaikka rivi 4 korvattaisiin rivillä **a++;**
- ☐ En halua vastata

Kysymys 1.6

Mitä break-lauseen jälkeen tapahtuu?

- ☐ Poistutaan silmukasta
- ☐ Keskeytetään ohjelman suoritus
- ☐ Palautetaan aliohjelman paluuarvo
- ☐ Jatketaan suoraan silmukan seuraavalle kierrokselle
- ☐ En halua vastata

Tehtävä 2 (6 p.)

Alla on monivalintakysymyksiä. Kuhunkin kysymykseen on täsmälleen yksi oikea vastaus. Arviointi:

- Oikea vastaus, +1 p.
- Väärä vastaus, -1 p.
- Ei vastattu, -1 p.
- Valittu “En halua vastata”, 0 p.

Tehtävän kokonaispistemäärä on mahdollisista miinusposteista huolimatta vähintään 0.

Kysymys 2.1

Mikä seuraavista väitteistä pitää paikkansa?

- ☐ Funktion dokumentaatiokommenteissa tulee kertoa parametrien nimet
- ☐ Dokumentaatiokommentit on tarkoitettu vain ohjelmoijalle itselleen
- ☐ Funktion dokumentaatiokommentit voi kirjoittaa vasta kun funktio on valmis
- ☐ Dokumentaatiokommentit kirjoitetaan funktion rajaavien aaltosulkujen väliin
- ☐ En halua vastata

Plugin jsrunner error:

Plugin does not support printing yet. Please refer to TIM help pages if you want to learn how you can manually define what to print here.

Kysymys 2.2

Oletetaan, että meillä on olemassa vain muuttujat `jonot` ja `etsittava`, jotka on alustettu seuraavasti:

```
List<string> jonot = new List<string>() { "kissa", "koira", "linnunpelätin" };  
string etsittava = "koira";
```

Oletetaan lisäksi, että meillä on funktio, jonka esittelyrivi on:

```
public static List<string> Suodata(List<string> jonot, string hakusana)
```

Mikä seuraavista on syntaktisesti oikea tapa kutsua funktiota Suodata?

- ☐ Suodata("kissa", "koira")
- ☐ jonot = Suodata(jonot, hakusana)
- ☐ Suodata(jonot, "etsittava")
- ☐ Suodata(jonot, etsittava) = uudetJonot
- ☐ En halua vastata

Kysymys 2.3

Mikä seuraavista on totta, jos haluamme, että ohjelmassa ei ole käänkösvirheitä?

- ☐ Ohjelmassa pitää olla rivinvaihtoja.
- ☐ Ohjelmassa pitää olla vähintään yksi using-lause.
- ☐ {-merkin jälkeen pitää tulla aina }-merkki ennen seuraavaa {-merkkiä.
- ☐ Avainsanojen välissä voi olla mielivaltaisen määrä välilyöntejä ja rivinvaihtoja.
- ☐ En halua vastata

Kysymys 2.4

Oletetaan, että meillä on kaksi sisäkkäistä for-silmukkaa. Mikä seuraavista pitää paikkansa?

- ☐ Sisempi silmukka suoritetaan aina vähintään yhden kerran.
- ☐ Kummassakin silmukassa on käytettävä saman nimisiä muuttujia.
- ☐ Ulomassa silmukassa voi olla sisemmän silmukan jälkeen suoritettavia lauseita.
- ☐ Sisemmässä silmukassa olevat lauseet suoritetaan ennen ulomman silmukan lauseita.
- ☐ En halua vastata

Kysymys 2.5

Alla on Funktio-niminen aliohjelma. Kysymykset 2.5 ja 2.6 liittyvät tähän aliohjelmaan.

```
public static int Funktio(int[] luvut)
{
    int yhteensa = 0;
    bool laskevaJono = false;
    for (int i = 1; i <= luvut.Length; i++)
    {
        if (i < luvut.Length && luvut[i] < luvut[i - 1])
        {
            laskevaJono = true;
        }
        else if (laskevaJono)
        {
            yhteensa++;
            laskevaJono = false;
        }
    }
    return yhteensa;
}
```

Mikä seuraavista kuvaa parhaiten aliohjelman toimintaa? Funktio palauttaa...

- ☐ laskevien osajonojen lukumäärän.
- ☐ negatiivisten lukujen lukumäärän.
- ☐ laskevien osajonojen yhteispituuden.
- ☐ tiedon, sisältääkö taulukko laskevia osajonoja vai ei.
- ☐ En halua vastata

Kysymys 2.6

Mikä seuraavista väitteistä on lähimpänä totuutta? Aliohjelma palauttaa arvon 0...

- ☐ vain silloin, kun taulukko on tyhjä.
- ☐ silloin, kun taulukko on tyhjä, tai taulukko sisältää vain samoja lukuja.
- ☐ silloin, kun taulukko sisältää vain laskevan osajonon, esimerkiksi [3, 2, 1].
- ☐ silloin, kun taulukko on tyhjä, tai taulukko sisältää vain negatiivisia lukuja.
- ☐ En halua vastata

Tehtävä 3 (6 p.)

Tee funktio `VaalinEkaKierros`. Funktio ottaa vastaan ehdokkaiden saamat äänimäärät kokonaislukutaulukkona. Funktio palauttaa kokonaislukutaulukon, jonka sisältö määräytyy seuraavasti:

- (a) jos voittaja sai **yli 50%** kaikista äänistä, taulukko sisältää vain voittajan äänimäärän,
- (b) muutoin taulukko sisältää kahden eniten ääniä saaneen ehdokkaan äänimäärät.

Tärkeää: Voit olettaa, että (i) taulukossa on vähintään yksi alkio, (ii) alkiot ovat positiivisia kokonaislukuja ja (iii) taulukko on järjestetty pienimmästä suurimpaan.

Alla on kaksi esimerkkiä funktion toiminnasta:

1. esimerkki: Suurin äänimäärä (60) ei ole enemmän kuin 50% kaikista äänistä.

```
int[] ehdokkaidenAanimaat = { 5, 10, 20, 40, 60 };
int[] toiselleKierrokselle = VaalinEkaKierros(ehdokkaidenAanimaat);
Console.WriteLine(string.Join(", ", toiselleKierrokselle)); // tulostaa "40, 60"
```

2. esimerkki: Suurin äänimäärä (80) on yli 50% kaikista äänistä.

```
int[] ehdokkaidenAanimaat = { 5, 10, 20, 40, 80 };
int[] toiselleKierrokselle = VaalinEkaKierros(ehdokkaidenAanimaat);
Console.WriteLine(string.Join(", ", toiselleKierrokselle)); // tulostaa "80"
```

Tehtävä 4 (6 p.)

Tee funktio `EriLuvut`, joka ottaa vastaan kaksi kokonaislukutaulukkoa. Funktio palauttaa uuden kokonaislukutaulukon, joka sisältää nousevassa suuruusjärjestyksessä kaikki ne luvut, jotka esiintyvät vain toisessa syötetaulukossa, mutta *eivät* molemmissa. Uusi taulukko saa sisältää lukuja vain kerran, vaikka ne esiintyisivät syötetaulukoissa useamman kerran.

Esimerkiksi seuraava pääohjelma


```
int[] luvut1 = [5, 5, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14];
int[] luvut2 = [1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 9, 10];
int[] yhteiset = EriLuvut(luvut1, luvut2);
Console.WriteLine(string.Join(", ", yhteiset));
```

tulostaisi 1, 2, 3, 4, 11, 12, 13, 14.

Tallenna palautuslaatikkoon vain funktio. (Korjaus tenttiin klo 13:07.)

Huomioi tentin alussa mainitut yleiset kriteerit ohjelmalle. Lisäksi ylimääräisistä if-lauseista tai muusta turhasta toistosta vähennetään pisteitä.

Huom! LINQ-funktioiden, kuten `Except` ja `Union` käyttö on kielletty.

Alla on muutamia vinkkejä, joita et välttämättä tarvitse, mutta ne voivat helpottaa tehtävän tekemistä.

Vinkki 1: Listan `Contains`-metodin avulla voi tutkia, sisältääkö lista jonkin alkion. Esimerkki:

```
List<int> luvut = new List<int> { 3, 2, 1 };
Console.WriteLine(luvut.Contains(3)); // True
Console.WriteLine(luvut.Contains(4)); // False
```

Vinkki 2: Listan `Sort`-metodin avulla listan voi järjestää nousevaan suuruusjärjestykseen. Äskeistä esimerkkiä jatkaen:

```
Console.WriteLine(string.Join(", ", luvut)); // 3, 2, 1
luvut.Sort();
Console.WriteLine(string.Join(", ", luvut)); // 1, 2, 3
```

Vinkki 3: Listan voi muuttaa taulukoksi `ToArray`-metodilla. Äskeistä esimerkkiä jatkaen:

```
int[] luvutTaulukkona = luvut.ToArray();
Console.WriteLine(string.Join(", ", luvutTaulukkona)); // 1, 2, 3
```

Tehtävät päättyvät tähän

Tästä linkistä pääset takaisin tentin johdantosivulle. Voit palata tälle sivulle tenttiajan puitteissa niin monta kertaa kuin haluat.

B Arviointikohteet

B.1 Tehtävä 3

```
Score = 1.0,  
Input = [3, 5, 8, 9, 15],  
Output = [9, 15],  
ErrorMessage = "Kun suurin äänimäärä ei ole yli 50% kaikista äänistä,  
palautetaan kaksi suurinta äänimäärää."
```

```
Score = 1.0,  
Input = [5, 5, 5, 5, 5],  
Output = [5, 5],  
ErrorMessage = "Jos ehdokkaat saavat saman äänimäärän,  
tulee palauttaa kaksi samaa lukua."
```

```
Score = 1.0,  
Input = [1, 2, 3, 4, 10],  
Output = [4, 10],  
ErrorMessage = "50% äänistä ei vielä riitä voittoon."
```

```
Score = 1.0,  
Input = [1, 2, 3, 4, 11],  
Output = [11],  
ErrorMessage = "Yli 50% äänistä riittää voittoon."
```

```
Score = 1.0,  
Input = [8, 8],  
Output = [8, 8],  
ErrorMessage = "Kun suurin äänimäärä ei ole yli 50%  
kaikista äänistä, palautetaan kaksi suurinta äänimäärää."
```

```
Score = 1.0,  
Input = [15],  
Output = [15],  
ErrorMessage = "Jos on vain yksi ehdokas, palautetaan se."
```

B.2 Tehtävä 4

```
Score = 1.0,  
Input1 = [1, 2, 3],  
Input2 = [4, 5, 6],  
Output = [1, 2, 3, 4, 5, 6],  
ErrorMessage = "Jos taulukot ovat täysin erilliset,  
tulee molempien sisältö mukaan tulostaulukkoon."
```

```
Score = 0.5,  
Input1 = [1, 1, 2],
```

```
Input2 = [1, 1, 3, 1, 1, 3],
Output = [2, 3],
ErrorMessage = "Jos taulukot sisältävät samoja alkioita,
ne eivät saa tulla tulostaulukkoon."
```

```
Score = 0.5,
Input1 = [1, 1, 2, 1, 1, 2],
Input2 = [1, 1, 3],
Output = [2, 3],
ErrorMessage = "Jos taulukot sisältävät samoja alkioita,
ne eivät saa tulla tulostaulukkoon."
```

```
Score = 1.0,
Input1 = [1, 1, 1],
Input2 = [1, 1, 1, 1, 1],
Output = [],
ErrorMessage = "Jos taulukot sisältävät vain samoja alkioita,
tuloslistan tulee olla tyhjä."
```

```
Score = 1.0,
Input1 = [4, 5, 6],
Input2 = [1, 2, 3],
Output = [1, 2, 3, 4, 5, 6],
ErrorMessage = "Tulostaulukon tulee olla nousevassa
suuruusjärjestyksessä."
```

```
Score = 1.0,
Input1 = [],
Input2 = [1, 1, 3, 1,],
Output = [1, 3],
ErrorMessage = "Toinen taulukoista voi olla tyhjä."
```

```
Score = 1.0,
Input1 = [],
Input2 = [],
Output = [],
ErrorMessage = "Molemmat taulukot voivat olla tyhjiä."
```