

# ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

22. toukokuuta 2024

## Yleistä

Tentti oli pistekeskiarvon 9,2 (keskihajonta 5,2) perusteella vaikea. Demohyvitysten kanssa keskiarvo oli 12,9. Huomaa, että demopisteet on laskettu tentin päälle, ja arvosana lasketaan vasta sen jälkeen. Opiskelijan omat tehtävät ovat nähtävissä TIMissä alkuperäisellä tenttisivulla. Uusintojen ajankohdat löydät Sisusta.

Taulukko 1: Tehtäviin vastausten antaneiden opiskelijoiden määrät ja keskiarvot.

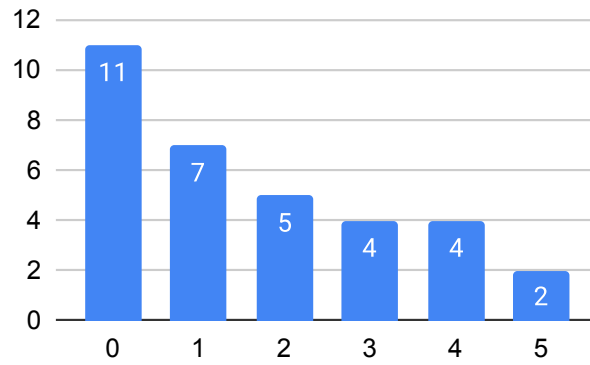
Tehtävä	Teki	Keskiarvo	Automaattinen arvostelu / Tarkastaja
T1	32	4,3	Antti-Jussi Lakanen
T2	32	2,7	Antti-Jussi Lakanen
T3	32	1,0	Antti-Jussi Lakanen
T4	32	1,5	Antti-Jussi Lakanen
<b>Yht</b>		<b>9,2</b>	(ennen demopisteitä)



JYVÄSKYLÄN YLIOPISTO

## Arvosteluasteikko ja arvosanjakauma

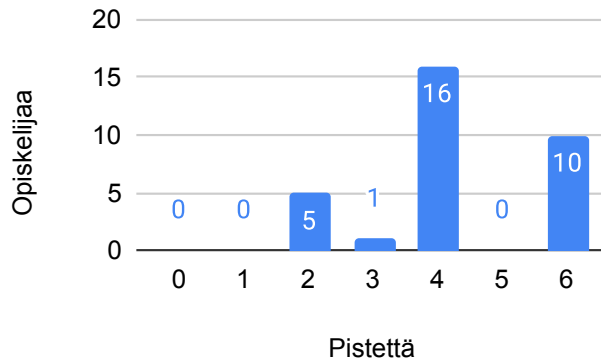
Arvolause	Pistemäärä (alaraja on inkluusiivinen, yläraja eksklusiivinen)
5	22
4	19
3	16
2	13
1	10



Tentin arvosanjakauma, kun demopisteet on laskettu tenttipisteisiin mukaan.

## Tehtävä 1 (6 p.)

1B 2A 3B 4A 5A 6C



Tehtävän 1 pistejakauma.

## Tehtävä 2 (6 p.)

Opiskelijan ratkaisua testattiin muun muassa seuraavilla syötteillä:

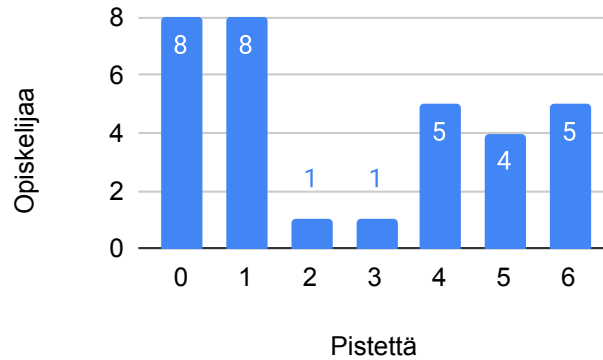
- Aineistossa on useita rivejä. Etsittäviä kirjaimia on useita, sekä suur- että pieni-kirjaimina.
- Aineistossa on useita rivejä, mutta vain yksi sana per rivi. Etsittäviä kirjaimia ei ole kaikilla riveillä.
- Aineisto on tyhjä.
- Aineistossa on yksi pitkä rivi, joka ei pääty rivinvaihtoon. Etsittävä merkki on annettu suuraakkosena; muissa testeissä etsittävä merkki on annettu pienaakkosena.

Jos kaikki testit menivät läpi, annettiin täydet pisteet. Jos jokin näistä testeistä ei mennyt läpi, vähennettiin tyypillisesti 2 pistettä. Jos virheitä oli useampi, vähennettiin pisteitä enemmän. Jos yksikään testi ei mennyt läpi, annettiin tyypillisesti 0 pistettä.

### Malliratkaisu

```
using System;
using System.IO;

/// @author Antti-Jussi Lakanen
/// @version 22.5.2024
/// <summary>
/// Tehtävä 2, lasketaan kirjaimia.
/// </summary>
public class T2_Kirjainlaskuri
```



Tehtävän 2 pistejakauma.

```

{
    /// <summary>
    /// Luetaan tiedosto ja kutsutaan funktiota.
    /// </summary>
    /// <param name="args">Ei käytössä.</param>
    public static void Main(string[] args)
    {
        string[] rivit = File.ReadAllLines("data.txt");
        int kirjaimia = MontakoKirjainta(rivit, 'a');
        Console.WriteLine(kirjaimia);
    }

    /// <summary>
    /// Funktio tutkii kuinka monta kertaa etsittävä merkki esiintyy aineistossa.
    /// Sekä suuri- että pienikirjaimet lasketaan samaksi merkiksi.
    /// </summary>
    /// <param name="rivit">Tutkittavat merkkijonot</param>
    /// <param name="merkki">Etsittävä merkki</param>
    /// <returns>Löytyneiden merkkien määrä</returns>
    public static int MontakoKirjainta(string[] rivit, char merkki)
    {
        int kirjaimia = 0;
        foreach (string rivi in rivit)
        {
            foreach (char c in rivi)
            {
                if (char.ToLower(c) == char.ToLower(merkki))
                {
                    kirjaimia++;
                }
            }
        }
        return kirjaimia;
    }
}

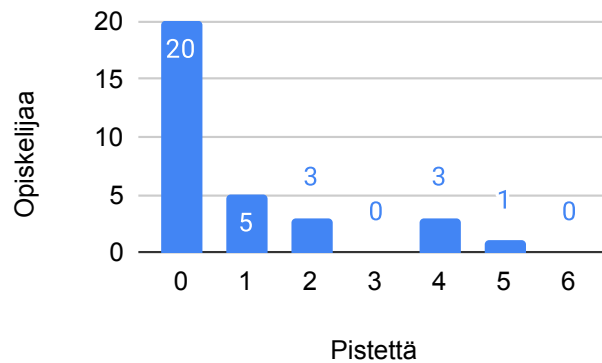
```

## Tehtävä 3 (6 p.)

Opiskelijan ratkaisua testattiin muun muassa seuraavilla syötteillä:

- [4, 2, 5, 1, 3, 3, 3, 2, 2, 8, 9, 10, 2, 3, 4]
- [4, 4]
- [4]
- [-1, 1, 0]
- `Array.Empty<int>()`

Jos kaikki testit menivät läpi, annettiin täydet pisteet. Jos jokin näistä testeistä ei mennyt läpi, vähennettiin tyypillisesti 2 pistettä. Jos virheitä oli useampi, vähennettiin pisteitä enemmän. Jos yksikään testi ei mennyt läpi, annettiin tyypillisesti 0 pistettä.



Tehtävän 3 pistejakauma.

### Malliratkaisu

Luokan ja pääohjelman dokumentaatiot on jätetty pois tilan säästämiseksi; katso vastaavat dokumentaatiot tehtävästä 2.

```
using System;
using System.Collections.Generic;

public class T3_PisinJono
{
    public static void Main(string[] args)
    {
        int[] luvut = [4, 2, 5, 1, 3, 3, 3, 2, 2, 8, 9, 10, 2, 3, 4];
        int[] pisinErilaiset = EtsiPisinErilaiset(luvut);
        Console.WriteLine(string.Join(", ", pisinErilaiset));
    }

    /// <summary>
```

```

/// Etsii pisinmän perättäisten lukujen joukon, jossa
/// kaikki luvut ovat keskenään erisuuria
/// </summary>
/// <param name="luvut">Luvut</param>
/// <returns>Pisin keskenään erilaiset</returns>
public static int[] EtsiPisinErilaiset(int[] luvut)
{
    int[] pisinJono = new int[0];
    List<int> nykyinenJono = new List<int>();
    for (int i = 0; i < luvut.Length; i++)
    {
        // Jos luku ei ole jo jonossa, lisätään se sinne
        if (!nykyinenJono.Contains(luvut[i]))
        {
            nykyinenJono.Add(luvut[i]);
            // Jos nykyinen jono on pisin tähän mennessä, päivitetään se
            if (nykyinenJono.Count > pisinJono.Length)
            {
                pisinJono = nykyinenJono.ToArray();
            }
        }
        // Jos luku on jo jonossa, aloitetaan uusi jono
        // Huomaa, että tässä kohtaa ei tarvitse tarkistaa,
        // onko nykyinen jono pisin koska jos se on,
        // se on jo päivitetty edellisellä kierroksella
        else
        {
            nykyinenJono = new List<int>();
        }
    }

    return pisinJono;
}
}

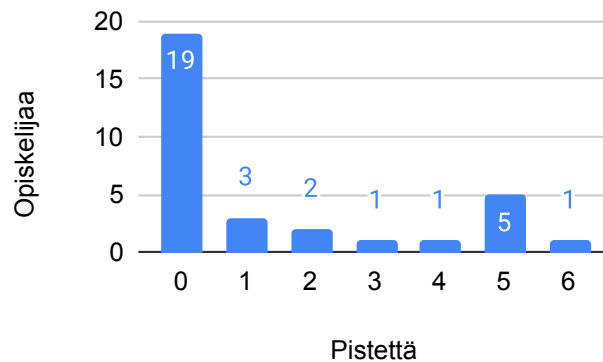
```

## Tehtävä 4 (6 p.)

Opiskelijan ratkaisua testattiin muun muassa seuraavilla syötteillä:

- [1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 9, 10],  
[5, 5, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] → [1]
- [1, 2, 3], [4, 5, 6] → []
- [1, 1, 2, 3], [1, 1, 1, 2, 2, 2, 3, 3, 3, 5, 5, 5] → [1, 2, 3]
- [1, 1, 2, 3], [] → []
- [], [] → []

Jos kaikki testit menivät läpi, annettiin täydet pisteet. Jos jokin näistä testeistä ei mennyt läpi, vähennettiin tyypillisesti 2 pistettä. Jos virheitä oli useampi, vähennettiin pisteitä enemmän. Jos yksikään testi ei mennyt läpi, annettiin tyypillisesti 0 pistettä.



Tehtävän 4 pistejakauma.

## Malliratkaisu

Luokan ja pääohjelman dokumentaatiot on jätetty pois tilan säästämiseksi; katso vastaavat dokumentaatiot tehtävästä 2.

```
using System;
using System.Collections.Generic;

public class T4_Yhteiset
{
    static void Main()
    {
        int[] luvut1 = [1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 9, 10];
        int[] luvut2 = [5, 5, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14];
        int[] yhteiset = YhteisetLuvut(luvut1, luvut2);
        Console.WriteLine(string.Join(", ", yhteiset));
        // '5, 6, 7, 8, 9, 10'.
    }

    /// <summary>
    /// Palauttaa uuden taulukon, joka sisältää kaikki ne luvut, jotka
    /// esiintyvät molemmissa syötetaulukoissa. Uusi taulukko sisältää
    /// lukuja vain kerran, vaikka ne esiintyisivät molemmissa
    /// syötetaulukoissa useamman kerran.
    /// </summary>
    /// <param name="luvut1">Luvut 1</param>
    /// <param name="luvut2">Luvut 2</param>
    /// <returns></returns>
    public static int[] YhteisetLuvut(int[] luvut1, int[] luvut2)
    {
```

```
List<int> yhteiset = new List<int>();
for (int i = 0; i < luvut1.Length; i++)
{
    for (int j = 0; j < luvut2.Length; j++)
    {
        if (luvut1[i] == luvut2[j] && !yhteiset.Contains(luvut1[i]))
        {
            yhteiset.Add(luvut1[i]);
        }
    }
}
return yhteiset.ToArray();
}
```

## A Kysymykset



# Ohjelmointi 1 (ITKP102), tentti 22.5.2024 klo 16:00-20:00

(Tästä linkistä pääset takaisin tentin johdantosivulle.)

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.

## Tehtävä 1

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.

### Kysymys 1.1

Kuinka monta eri arvoa bool-tyyppinen muuttuja voi saada?

- 1
- 2
- 4
- 8
- En halua vastata

### Kysymys 1.2

Mikä seuraavista aiheuttaa käänkövirheen?

- Numero aliohjelman nimen alussa
- Numero aliohjelman nimen lopussa
- Suuraakkos-skandi (Å/Ä/Ö) aliohjelman nimen alussa
- Kokonaan pienaakkosilla kirjoitettu aliohjelman nimi
- En halua vastata

### Kysymys 1.3

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

- Ulommassa silmukassa *ei* saa olla return-lausetta
- Sisempää silmukkaa *ei* välttämättä suoriteta yhtään kierrosta
- Kummankin silmukan määrittelyrivillä voidaan määritellä i-niminen muuttuja
- Kummassakin silmukassa on käytettävä samannimistä muuttujaa silmukan määrittelyrivillä
- En halua vastata

### Kysymys 1.4

Mitä break-lauseen jälkeen tapahtuu?

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

### Kysymys 1.5

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

- Koko ohjelman voi kirjoittaa ilman ainuttakaan rivinvaihtoa
- Rivinvaihdon saa laittaa mihin vain, vaikka keskelle avainsanaa
- {-merkin jälkeen (mm. aliohjelmat ja silmukat) pitää tulla aina rivinvaihto
- Jokainen puolipisteeseen (;-merkki) päättyvä rivi tulee olla omalla rivillään
- En halua vastata

## Kysymys 1.6

Mikä seuraavista aiheuttaa käänkösvirheen?

- Ohjelmassa *ei* ole yhtään tulostuslausetta
- Ohjelmassa on alustettu muuttuja, jota ei käytetä
- Ohjelmassa käytetään muuttujaa, jota ei ole alustettu
- Ohjelmassa kutsutaan funktiota, jonka paluuarvoa ei käytetä mihinkään
- En halua vastata

## Tehtävä 2 (6 p.)

Tee ohjelma, joka lukee tekstitiedoston merkkijonotaulukoksi. Pääohjelmassa voidaan kutsua `MontakoKirjainta`-funktioita. Funktiolle annetaan kaksi argumenttia: annetaan argumentteina äsken luotu taulukko sekä jokin etsittävä merkki (ks. esimerkki alla). Funktion tulee palauttaa, kuinka monta kertaa etsittävä merkki esiintyy aineistossa. Sekä suuri- että pienikirjaimet lasketaan samaksi merkiksi.

Esimerkiksi jos tiedoston sisältö on

```
Kissa kävelee Alajärven  
kaduilla. Se ihmettelee  
maailman menoa.
```

niin seuraava pääohjelma

```
string[] rivit = File.ReadAllLines("data.txt");  
int kirjaimia = MontakoKirjainta(rivit, 'a');  
Console.WriteLine(kirjaimia);
```

tulostaisi 9.

Tallenna palautuslaatikkoon ohjelma kokonaisuudessaan (using-lauseet, pääohjelma, funktio). Huomioi tentin alussa mainitut yleiset kriteerit ohjelmalle. Lisäksi ylimääräisistä if-lauseista tai muusta turhasta toistosta vähennetään pisteitä.

Tehtävässä ei ole automaattista arvostelua, vaan se arvioidaan käsin.

## Tehtävä 3 (6 p.)

Tee funktio, joka ottaa vastaan kokonaislukutaulukon. Funktio etsii pisimmän perättäisten lukujen joukon, jossa kaikki luvut ovat keskenään erisuuria. Funktio palauttaa tämän lukujen joukon taulukkona. Esimerkiksi seuraava pääohjelma

```
int[] luvut = [4, 2, 5, 1, 3, 3, 3, 2, 2, 8, 9, 10, 2, 3, 4];
int[] pisinErilaiset = EtsiPisinErilaiset(luvut);
Console.WriteLine(string.Join(", ", pisinErilaiset));
```

tulostaisi '8, 9, 10, 2, 3, 4'.

Tallenna palautuslaatikkoon ohjelma kokonaisuudessaan (using-lauseet, pääohjelma, funktio). Huomioi tentin alussa mainitut yleiset kriteerit ohjelmalle. Lisäksi ylimääräisistä if-lauseista tai muusta turhasta toistosta vähennetään pisteitä.

Tehtävässä ei ole automaattista arvostelua, vaan se arvioidaan käsin.

## Tehtävä 4 (6 p.)

Tee funktio, joka ottaa vastaan kaksi kokonaislukutaulukkoa. Funktio palauttaa uuden taulukon, joka sisältää kaikki ne luvut, jotka esiintyvät molemmissa syötetaulukoissa. Uusi taulukko saa sisältää lukuja vain kerran, vaikka ne esiintyisivät molemmissa syötetaulukoissa useamman kerran.

Esimerkiksi seuraava pääohjelma

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

tulostaisi '5, 6, 7, 8, 9, 10'.

Tallenna palautuslaatikkoon ohjelma kokonaisuudessaan (using-lauseet, pääohjelma, funktio). Huomioi tentin alussa mainitut yleiset kriteerit ohjelmalle. Lisäksi ylimääräisistä if-lauseista tai muusta turhasta toistosta vähennetään pisteitä.

**Huom!** LINQ-funktioiden, kuten `Intersect` käyttö on kielletty.

Tehtävässä ei ole automaattista arvostelua, vaan se arvioidaan käsin.

## Tehtävät päättyvät tähän

(Tästä linkistä pääset takaisin tentin johdantositulle.)