

# ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

29. huhtikuuta 2026

## Yleistä

Tentti oli pistekeskiarvon 10,7 (keskihajonta 6,5) perusteella vaikeudeltaan vaikeahko. Demohyvitysten kanssa keskiarvo oli 14,9. Huomaa, että demopisteet on laskettu tentistä saatujen pisteiden päälle, ja arvosana lasketaan vasta sen jälkeen. Opiskelijan omat tehtävät ovat nähtävissä TIMissä alkuperäisellä tenttisivulla. Uusintojen ajankohdat löydät kurssin kotisivulta.

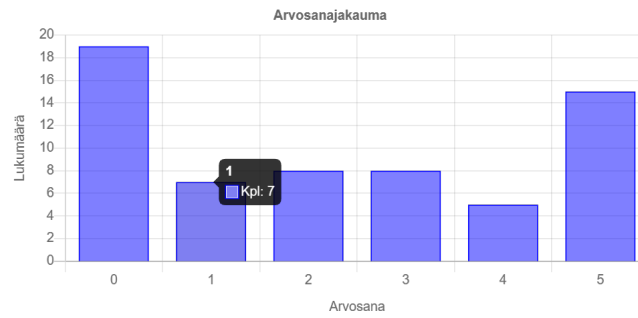
Tehtävä	Teki	Keskiarvo
T1	62	4,2
T2	62	4,4
T3	51	1,5
T4	40	1,4
<b>Yht (ennen demopisteitä)</b>		<b>10,7</b>



JYVÄSKYLÄN YLIOPISTO

# Arvosteluasteikko ja arvosanjakauma

Pistemäärä (inklusiivinen alaraja)	Arvosana
11	1
13	2
15	3
17	4
19	5



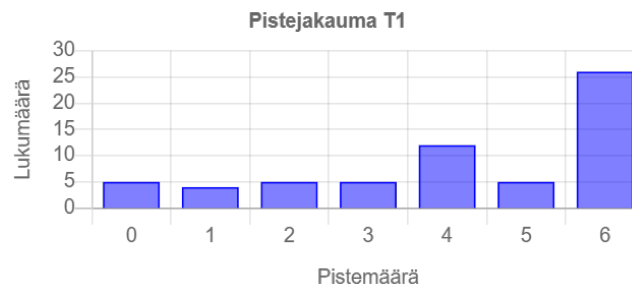
Tentin arvosanjakauma.

# Tehtävä 1 (6 p.)

Oikeat vastaukset on lihavoitu.

- Kysymys 1.1: Mikä seuraavista väitteistä pitää paikkansa?
  - Muuttujan arvoa voi muuttaa täsmälleen yhdessä lauseessa.
  - 'Kokonaislukumuuttujaan (int) voi sijoittaa desimaaliluvun, jolloin se pyöristyy automaattisesti lähimpään kokonaislukuun.'
  - **Ohjelman suorituksen aikana muuttujan arvoa voidaan muuttaa sijotuslauseella, jolloin vanha arvo korvautuu uudella.**
  - Muuttuja on esiteltävä (engl. \*declare\*) uudelleen tyyppineen jokaisessa silmukan vaiheessa, jos sen arvoa halutaan muuttaa.
  - En halua vastata
- Kysymys 1.2: Mikä seuraavista väitteistä pitää paikkansa?
  - Taulukon ensimmäinen alkio on aina indeksissä 1.
  - Taulukon alkioita voi poistaa siten, että taulukon koko (Length) pienenee automaattisesti.
  - **Taulukon viimeisen alkion indeksi on yhtä pienempi kuin taulukon pituus.**
  - Kokonaislukutaulukkoon (int[]) voi tallentaa myös merkkijonoja, jos ne sisältävät vain numeroita.
  - En halua vastata
- Kysymys 1.3: Mikä seuraavista väitteistä pitää paikkansa aliohjelman paluuarvosta?
  - Paluuarvo tarkoittaa sitä, että aliohjelma tulostaa (esim. Console.WriteLine) tiedon käyttäjälle näytölle.
  - void-tyyppinen aliohjelma palauttaa aina arvon 0 tai tyhjän merkkijonon.
  - Aliohjelmassa olevan return-lauseen jälkeen kirjoitetut lauseet suoritetaan aina ennen aliohjelmasta poistumista.
  - **Kun aliohjelma suorittaa return-lauseen, sen suoritus päättyy ja se palauttaa annetun arvon aliohjelman kutsujalle.**
  - En halua vastata
- Kysymys 1.4: Mikä seuraavista väitteistä pitää paikkansa?
  - if-lauseetta seuraava else-lohko on pakollinen kirjoittaa joka kerta.
  - Samassa ehtolauseketjussa voi olla korkeintaan yksi else if -lohko.
  - **if-lauseen ehtolauseke on aina totuusarvotyyppinen**
  - Jos if-lauseen ehto on epätosi, ohjelman suoritus keskeytyy ja antaa virheilmoituksen.

- En halua vastata
- Kysymys 1.5: Mikä seuraavista väitteistä pitää paikkansa?
  - for-silmukan toistokertojen määrä tulee aina riippua muuttujasta.
  - while-silmukkaa voi käyttää ainoastaan taulukoiden läpikäymiseen.
  - Toistolause lopettaa koko ohjelman suorituksen.
  - **Toistolauseen suoritus voi päättyä, vaikka silmukan ehtolauseke olisi edelleen tosi.**
  - En halua vastata
- Kysymys 1.6: Mikä seuraavista väitteistä pitää paikkansa parametreista?
  - Aliohjelmaa kutsuttaessa argumenttina annettavan muuttujan nimen on oltava täsmälleen sama kuin aliohjelman esittelyrivillä määritelty parametrin nimi.
  - Parametrin arvon muuttaminen aliohjelman sisällä muuttaa automaattisesti myös aliohjelman kutsussa olevan muuttujan arvoa.
  - Aliohjelmalla on aina oltava vähintään yksi parametri, jotta sitä voidaan kutsua.
  - **Parametrien avulla aliohjelmalle voidaan välittää tietoa kutsuvasta ohjelmasta, jotta samaa aliohjelmaa voidaan hyödyntää eri arvoilla.**
  - En halua vastata



Kuva 1: Tehtävän 1 pistejakauma.

## Tehtävä 2 (6 p.)

Oikeat vastaukset on lihavoitu.

Kuhunkin kysymykseen on täsmälleen yksi oikea vastaus. Arviointi:

- Oikea vastaus, 2 p.
- Väärä vastaus, -1 p.
- Ei vastattu, -1 p.

- Valittu "En halua vastata", 0 p.

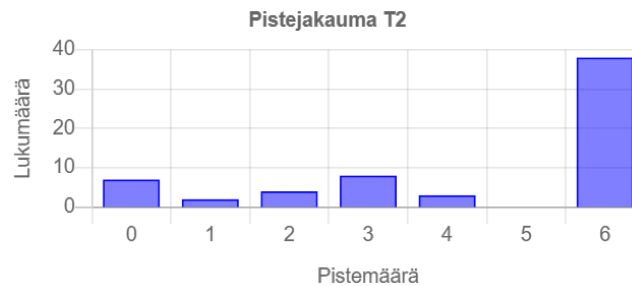
Tehtävän minimipistemäärä on 0 p.

Alla on aliohjelma X ja kolme sitä koskevaa kysymystä. Funktion ja muuttujien nimet on tehtävän luonteesta johtuen kirjoitettu lyhykäisesti.

```
public static int X(int[] t)
{
    if (t.Length == 0) return 0;
    int a = 1;
    int b = 1;
    for (int j = 1; j < t.Length; j++)
    {
        if (t[j] == t[j - 1]) b++;
        else
        {
            if (b > a) a = b;
            b = 1;
        }
    }
    if (b > a) a = b;
    return a;
}
```

- Kysymys 2.1: Mikä oheisista kuvauksista parhaiten sopii dokumentoimaan X-aliohjelman toimintaa? X-aliohjelma...
  - palauttaa taulukon suurimman alkion arvon.
  - palauttaa, montako eri lukua taulukossa esiintyy.
  - palauttaa viimeisen peräkkäisten samojen alkioden jakson pituuden.
  - **palauttaa pisimmän peräkkäisten yhtäsuurten alkioden jakson pituuden.**
  - En halua vastata
- Kysymys 2.2: Mikä seuraavista väitteistä pitää paikkansa?
  - Aliohjelma palauttaa tyhjälle taulukolle 1.
  - Aliohjelma palauttaa aina taulukon ensimmäisen alkion arvon.
  - Jos taulukko sisältää negatiivisia lukuja, aliohjelma palauttaa aina 0.
  - **Jos taulukossa ei ole kahta vierekkäistä yhtäsuurta alkioita, aliohjelma palauttaa 1.**
  - En halua vastata
- Kysymys 2.3: Miksi ennen return-lausetta tehdään vielä tarkistus if (b > a) a = b;?
  - Jotta taulukon pituus voidaan tallentaa muuttujaan a.

- Jotta taulukon ensimmäinen alkio ei jäisi laskematta mukaan.
- Jotta kaikki negatiiviset luvut muutetaan positiivisiksi ennen palautusta.
- **Jotta myös taulukon lopussa oleva jakso tulee huomioiduksi, vaikka sil-  
mukka ei enää kohtaa eri arvoa.**
- En halua vastata



Tehtävän 2 pistejakauma.

## Tehtävä 3 (6 p.)

Tee funktio `SuunnanMuutokset`, joka ottaa parametrina `int`-taulukon ja palauttaa, montako kertaa peräkkäisten alkioiden välinen muutos vaihtaa suuntaa.

Suunnan muutos tarkoittaa sitä, että ensin ollaan siirtymässä kasvavaan suuntaan ja myöhemmin laskevaan suuntaan, tai päinvastoin. Peräkkäiset samat arvot eivät it-sessään muuta suuntaa, ja ne voidaan ohittaa suunnan muutosta laskettaessa.

**Esimerkki 1:** Syöte: `luvut = [1, 3, 5, 4, 2, 6]`, paluuarvo: 2. **Selitys:** Ensin suunta on kasvava (1 -> 3 -> 5), sitten laskeva (5 -> 4 -> 2) ja lopuksi taas kasvava (2 -> 6). Suunta muuttuu siis kaksi kertaa.

**Esimerkki 2:** Syöte: `luvut = [2, 2, 3, 5, 5, 4]`, paluuarvo: 1. **Selitys:** Suunta on ensin kasvava (2 -> 2 -> 3 -> 5 -> 5) ja lopuksi laskeva (5 -> 4), joten suunnan muutos tapahtuu kerran. Peräkkäiset samat arvot (2 -> 2 ja 5 -> 5) eivät muuta suuntaa, joten ne ohitetaan.

Jos taulukossa on alle kaksi alkioita, funktion tulee palauttaa 0.

### Malliratkaisu

```

/// <summary>
/// Laskee, montako kertaa peräkkäisten alkioiden välinen muutos
/// vaihtaa suuntaa.
/// </summary>
/// <param name="luvut">Taulukko, jonka suunnan muutokset lasketaan</param>
/// <returns>Suunnan muutosten määrä</returns>
public static int SuunnanMuutoksetMALLI(int[] luvut)
{
    if (luvut.Length < 2) return 0;

```

```

var muutokset = 0;
var edellinenSuunta = 0; // 1 = kasvava, -1 = laskeva

for (var i = 1; i < luvut.Length; i++)
{
    var suunta = 0;

    if (luvut[i] > luvut[i - 1]) suunta = 1;
    else if (luvut[i] < luvut[i - 1]) suunta = -1;
    else continue;

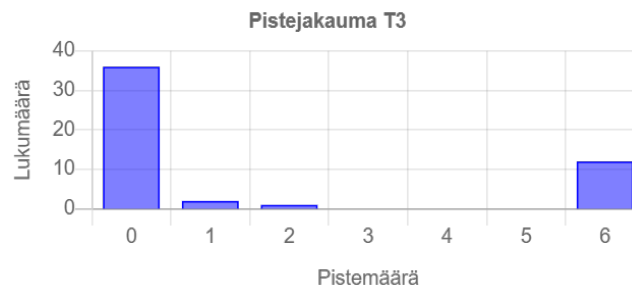
    if (edellinenSuunta != 0 && suunta != edellinenSuunta)
    {
        muutokset++;
    }

    edellinenSuunta = suunta;
}

return muutokset;
}

```

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



Tehtävän 3 pistejakauma.

## Tehtävä 4 (6 p.)

Tee funktio `TasapainoIndeksi`. Funktio ottaa parametrina `int`-taulukon luvut.

`TasapainoIndeksi`llä tarkoitetaan indeksiä  $i$ , jonka vasemmalla puolella olevien alkioiden summa on yhtä suuri kuin oikealla puolella olevien alkioiden summa. Funktion tulee palauttaa pienin tällainen indeksi. Jos sopivaa indeksiä ei ole, funktion tulee palauttaa  $-1$ .

Huomaa, että taulukon ensimmäisen alkion vasemmalla puolella ei ole alkioita, joten vasemman puolen summa on silloin  $0$ . Vastaavasti viimeisen alkion oikean puolen summa voi olla  $0$ .

**Esimerkki 1:** Syöte: luvut = [1, 7, 3, 6, 5, 6], paluuarvo: 3. **Selitys:** Indeksien 3 vasemmalla puolella olevien lukujen summa on  $1 + 7 + 3 = 11$  ja oikealla puolella olevien lukujen summa on  $5 + 6 = 11$ .

**Esimerkki 2:** Syöte: luvut = [1, 2, 3], paluuarvo: -1. **Selitys:** Taulukossa ei ole indeksiä, jossa vasemman ja oikean puolen summat olisivat samat.

Voit olettaa, että luvut.Length on vähintään 1.

Jos taulukossa on useita tasapainoindeksijä, funktion tulee palauttaa niistä pienin.

Arvostelu pohjautui automaattiseen arvioon Liitteessä A Taulukossa A.2 esitettyjen arvostelukohteiden/testitapausten kautta. Automaattisilla arvostelukohteilla on tärkeysjärjestys: Mikäli ratkaisu ei toteuta ylempänä olevaa kohdetta, ei sitä alempia arvostelukohteita oteta huomioon. Pisteet pyöristettiin alaspäin lähimpään kokonaislukuun.

## Malliratkaisu

```
/// <summary>
/// Laskee taulukosta pienimmän indeksin, jonka vasemmalla
/// ja oikealla puolella olevien
/// alkioden summa on sama.
/// </summary>
/// <param name="luvut">Taulukko, josta tasapainoindeksi lasketaan</param>
/// <returns>Pienin tasapainoindeksi tai -1, jos sellaista ei ole</returns>
public static int TasapainoIndeksi(int[] luvut)
{
    int kokonaisSumma = 0;
    foreach (int luku in luvut) kokonaisSumma += luku;

    int vasenSumma = 0;

    for (int i = 0; i < luvut.Length; i++)
    {
        int oikeaSumma = kokonaisSumma - vasenSumma - luvut[i];
        if (vasenSumma == oikeaSumma) return i;
        vasenSumma += luvut[i];
    }

    return -1;
}
```



Tehtävän 4 pistejakauma.

# A Arviointikohteet

## A.1 Tehtävä 3

#	Syöte	Tulos	Pisteet	Virheilmoitus
1	Satunnainen	0	0.5	Funktio ei toimi satunnaisella syötteellä
2	[3, 1, 4]	1	0.5	Testaa toisen suunnanmuutoksen perusmuodon: ensin lasku, sitten kasvu.
3	[1, 3, 5, 4, 2, 6]	2	0.5	Testaa useamman suunnanmuutoksen ketjun. Tässä mennään kasvavasta laskevaan ja lopuksi taas kasvavaan.
4	[2, 2, 3, 5, 5, 4]	1	0.5	Testaa, että peräkkäiset samat arvot ohitetaan oikein. Varsinainen suunta on kasvu ja sen jälkeen lasku.
5	[1, 1, 1, 1]	0	0.5	Testaa tapauksen, jossa kaikki alkiot ovat samoja. Koska todellista suuntaa ei muodostu lainkaan, muutoksia ei tule.
6	[]	0	0.5	Testaa tyhjän taulukon. Alle kahden alkion taulukossa suunnanmuutoksia ei voi olla.
7	[7]	0	0.5	Testaa yhden alkion taulukon. Yksittäisestä alkioista ei synny nousua eikä laskua.
8	[1, 2]	0	0.5	Testaa kahden alkion kasvavan tapauksen. Suunta kyllä muodostuu, mutta ei vielä ehdi vaihtua.
9	[5, 3]	0	0.5	Testaa kahden alkion laskevan tapauksen. Myöskään tässä suunnanvaihtoa ei voi vielä tapahtua.
10	[1, 2, 3, 4]	0	0.5	Testaa tasaisesti kasvavan taulukon. Suunta pysyy koko ajan samana.

#	Syöte	Tulos	Pisteet	Virheilmoitus
11	[4, 3, 2, 1]	0	0.5	Testaa tasaisesti laskevan taulukon. Suunta pysyy koko ajan samana.
12	[1, 2, 2, 2, 1, 1, 3, 3, 2]	3	0.5	Testaa yhdistelmätilanteen, jossa mukana on sekä nousuja, laskuja että toistuvia samoja arvoja. Samat arvot pitää sivuuttaa, jolloin suunnat ovat kasvu, lasku, kasvu, lasku ja muutoksia kertyy kolme.

## A.2 Tehtävä 4

#	Syöte	Tulos	Pisteet	Virheilmoitus
1	[7]	0	0.5	Testaa yhden alkion taulukon. Ainoan alkion vasemmalla ja oikealla puolella summa on 0, joten tasapainoindeksi on 0.
2	[1, 7, 3, 6, 5, 6]	3	0.5	Testaa tehtävän ensimmäisen esimerkin. Tasapainoindeksi löytyy keskeltä taulukkoa.
3	[1, 2, 3]	-1	0.5	Testaa tehtävän toisen esimerkin. Taulukossa ei ole tasapainoindeksiä.
4	[2, 0, 2]	1	0.5	Testaa yksinkertaisen tapauksen, jossa tasapainoindeksi on indeksissä 1. Nykyistä alkiota ei lasketa mukaan kumpaankaan summaan.
5	[5, -2, 2, 0]	0	0.5	Testaa, että ensimmäinen indeksi voidaan hyväksyä. Vasemman puolen summa on 0 ja oikean puolen summa on myös 0.

#	Syöte	Tulos	Pisteet	Virheilmoitus
6	[1, -1, 3]	2	0.5	Testaa, että viimeinen indeksi voidaan hyväksyä. Oikean puolen summa on 0 ja vasemman puolen summa on myös 0.
7	[0, 0, 0]	0	0.5	Testaa taulukon, jossa on useita mahdollisia tasapainoindeksejä. Funktion tulee palauttaa niistä pienin.
8	[-2, -1, 3, -1, -2]	2	0.5	Testaa negatiivisia lukuja sisältävän taulukon. Tasapaino voi syntyä myös negatiivisista summista.
9	[2, -1, 1, 2]	-1	0.5	Testaa tapausta, jossa tasapainoindeksiä ei ole, vaikka taulukossa on sekä positiivisia että negatiivisia lukuja.
10	[-1, 0, -1]	1	0.5	Testaa, että nolla nykyisenä alkiona käsitellään oikein eikä sitä lasketa vasempaan tai oikeaan summaan.
11	[4, 1, -5, 2, 0, 2]	0	0.5	Testaa pidemmän taulukon, jossa tasapainoindeksi löytyy vasta myöhemmin. Tämä paljastaa virheitä, joissa summia päivitetään väärässä kohdassa.
12	[10, -3, 2, -8, 4]	-1	0.5	Testaa tapausta, jossa kokonaisumman parillisuus tai nykyisen alkion arvo ei yksin riitä ratkaisuun. Jokaisen indeksin vasen ja oikea summa pitää tarkistaa.