

# ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

10. huhtikuuta 2026

## Yleistä

Tentti oli pistekeskiarvon 12,1 (keskihajonta 6,8) perusteella vaikeudeltaan keskitasoa. Demohyvitysten kanssa keskiarvo oli 16,8. 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.

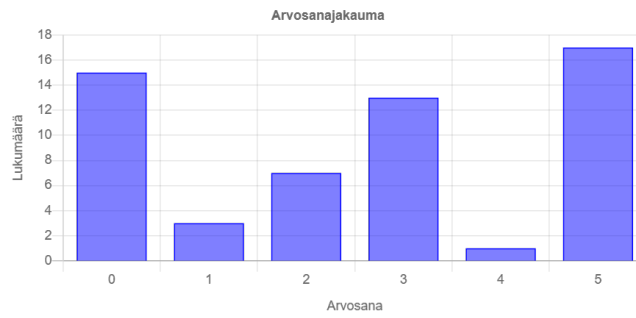
Tehtävä	Teki	Keskiarvo
T1	56	4,2
T2	56	4,0
T3	47	2,8
T4	46	1,9
<b>Yht (ennen demopisteitä)</b>		<b>12,1</b>



JYVÄSKYLÄN YLIOPISTO

# Arvosteluasteikko ja arvosanjakauma

Pistemäärä (inklusiivinen alaraja)	Arvosana
11	1
13	2
16	3
20	4
23	5



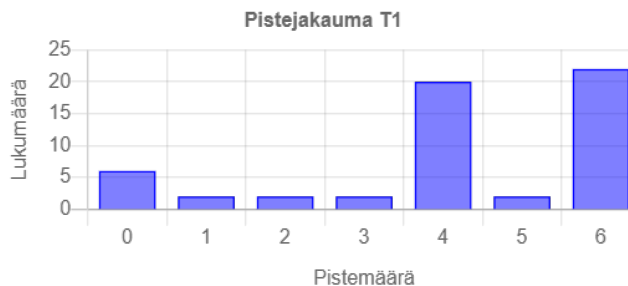
Tentin arvosanjakauma.

## Tehtävä 1 (6 p.)

Oikeat vastaukset on lihavoitu.

- Kysymys 1.1: Mikä seuraavista väitteistä pitää paikkansa? Muuttuja on...
  - aliohjelman osa, joka suoritetaan, kun sitä kutsutaan.
  - tapa tulostaa ruudulle tietoa helposti luettavassa muodossa.
  - **symboli, joka viittaa tietokoneen muistiin tallennettuun tietoon.**
  - ohjelmointikielen ominaisuus, jonka avulla ohjelma voi lukea tietoa käyttäjältä.
- Kysymys 1.2: Mikä seuraavista väitteistä pitää paikkansa?
  - Aliohjelman määrittelyssä tulee aina käyttää parametreja.
  - Aliohjelma suoritetaan aina vähintään kerran ohjelman aikana.
  - Aliohjelman suorituksen jälkeen koko ohjelman suoritus päättyy.
  - **Aliohjelmassa määritellään, mitä tapahtuu, kun aliohjelmaa kutsutaan.**
- Kysymys 1.3: Mikä seuraavista väitteistä pitää paikkansa?
  - Ehtolause voidaan korvata toistolauseella.
  - **Ehtolauseella voidaan toteuttaa joko-tai-rakenne.**
  - Ehtolauseessa tulee olla if-, else if-, ja else-haarat.
  - Ehtolause tulee aina olla ohjelman pääohjelmassa, ei aliohjelmissa.
- Kysymys 1.4: Mikä seuraavista väitteistä pitää paikkansa?
  - Toistolause suoritetaan aina vähintään yhden kerran.
  - Toistolauseessa tulee aina olla vähintään yksi ehtolause.
  - **Toistolauseessa voi olla määritettynä useita lopetusehtoja.**
  - Toistolauseessa tulee aina olla määritettynä montako kertaa toistolause suoritetaan.
- Kysymys 1.5: Mikä seuraavista väitteistä pitää paikkansa?
  - Sijoituslause voidaan tehdä vain ohjelman pääohjelmassa.
  - Sijoituslauseessa verrataan, ovatko vasen ja oikea puoli samat.
  - Sijoituslauseessa tulee aina olla aritmeettinen laskutoimitus tai ohjelmakutsu.
  - **Sijoituslauseessa oikean puolen arvo tallennetaan vasemman puolen osoittamaan kohteeseen.**
- Kysymys 1.6: Mikä seuraavista väitteistä pitää paikkansa?

- Lauseke voi sisältää aliohjelmakutsun.
- Lausekkeessa tulee aina olla aritmeettinen laskutoimitus.
- Lauseke on ohjelman osa, joka suoritetaan, kun sitä kutsutaan.
- Lauseke, muuttuja ja vakio ovat kaikki samoja asioita, mutta eri nimillä.



Kuva 1: Tehtävän 1 pistejakauma.

## Tehtävä 2 (6 p.)

Oikeat vastaukset on lihavoitu.

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

```
public static double Funktio(string x)
{
    if (x.Length == 0) return 0;
    char[] n = new char[] { '.', ',', ' ', '!' };
    int k = x.Length;
    foreach (char t in x)
    {
        if (n.IndexOf(t) >= 0) k--;
    }
    return 1.0 * k / x.Length;
}
```

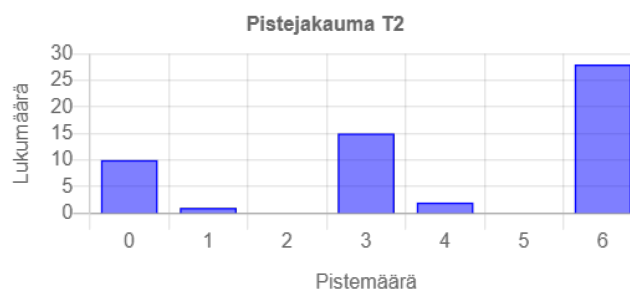
Kuhunkin kysymykseen on täsmälleen yksi oikea vastaus. Arviointi:

- Oikea vastaus, 2 p.
- Väärä vastaus, -1 p.
- Ei vastattu, -1 p.
- Valittu "En halua vastata", 0 p.

Tehtävän minimipistemäärä on 0 p.

- Kysymys 2.1: Mikä oheisista kuvauksista parhaiten sopii dokumentoimaan Funktio-aliohjelman toimintaa? Funktio-aliohjelma...

- palauttaa valikoitujen erikoismerkkien osuuden merkkijonon pituudesta.
  - palauttaa merkkijonossa olevien muiden kuin valikoitujen erikoismerkkien määrän.
  - **palauttaa muiden kuin valikoitujen erikoismerkkien osuuden merkkijonon pituudesta.**
  - laskee montako kertaa merkkijonossa esiintyy pisteitä, pilkkuja, välilyöntejä ja huutomerkkejä.
- Kysymys 2.2: Mikä seuraavista väitteistä pitää paikkansa?
    - Jos merkkijono on tyhjä, ohjelma kaatuu.
    - **Funktio palauttaa aina arvon välillä [0, 1].**
    - Jos merkkijonossa ei ole lainkaan valikoituja erikoismerkkejä, Funktio palauttaa arvon 0.
    - Funktion palauttama arvo on aina vähintään yhtä suuri kuin merkkijonossa olevien muiden kuin valikoitujen erikoismerkkien määrä.
  - Kysymys 2.3: Miksi viimeisellä rivillä on kerrottu luvulla 1.0?
    - Sitä ei tarvita - lasku toimisi oikein ilman sitäkin.
    - C#-kielessä int-tyyppisten muuttujien jakolasku on syntaksivirhe.
    - Sillä varmistetaan, että lasku suoritetaan kokonaislukulaskuna, jolloin saadaan tarkka kokonaisluku.
    - **Sillä varmistetaan, että lasku suoritetaan liukulukulaskuna, jolloin saadaan mahdollisimman tarkka tulos desimaalilukuna.**

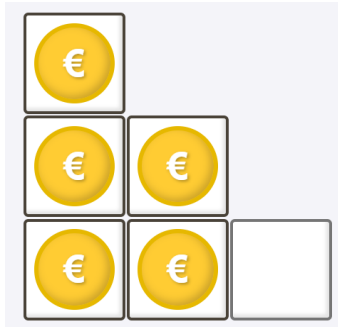


Tehtävän 2 pistejakauma.

## Tehtävä 3 (6 p.)

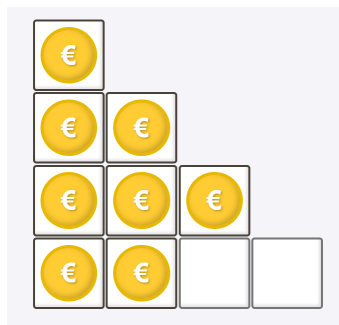
Sinulla on  $n$  kpl kolikoita, ja haluat rakentaa niistä portaikon. Portaikko koostuu  $k$  rivistä siten, että  $i$ :nnellä rivillä on tasan  $i$  kolikkoa. Portaikon viimeinen rivi voi kuitenkin olla vajaa.

Tee funktio 'Riveja', joka ottaa kokonaisluvun  $n$  ja palauttaa niiden \*täysien rivien\* määrän, jotka portaikkoon muodostuvat.



**Esimerkki 1:** Syöte:  $n = 5$ , paluarvo: 2. **Selitys:** Koska kolmas rivi jää vajaaksi (siihen tarvittaisiin 3 kolikkoa, mutta jäljellä on vain 2), palautetaan 2. Kuva:

**Esimerkki 2:** Syöte:  $n = 8$ , paluarvo: 3. **Selitys:** Koska neljäs rivi jää vajaaksi, palautetaan 3. Kuva:



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.

### Malliratkaisuja (2 eri versiota)

```

/// <summary>
/// Laskee, montako täyttä riviä portaikkoon muodostuu,
/// kun käytössä on n kolikkoa.
/// </summary>
/// <param name="n">Kolikoiden määrä</param>
/// <returns>Portaikoon täysien rivien määrä</returns>
public static int Riveja(int n)
{
    int i = 0;
    if (n < 0) return 0;
    while (true)
    {
        int summa = AritmeettinenSumma(i);
        if (summa > n) return i - 1;
    }
}

```

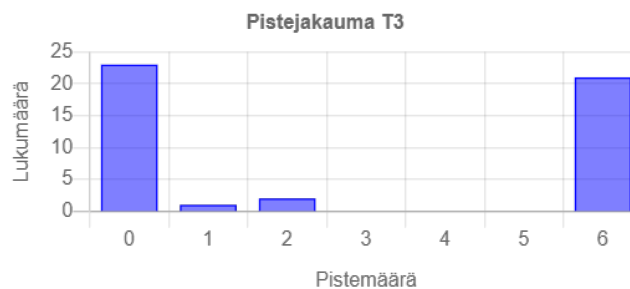
```

        i++;
    }
}

/// <summary>
/// Laskee, montako täyttä riviä portaikkoon muodostuu,
/// kun käytössä on n kolikkoa.
/// </summary>
/// <param name="n">Kolikoiden määrä</param>
/// <returns>Portaikon täysien rivien määrä</returns>
public static int Riveja2(int n)
{
    int riveja = 0;
    int kolikoitaKaytetty = 0;
    if (n < 0) return 0;
    while (true)
    {
        ++riveja;
        kolikoitaKaytetty += riveja;
        if (n < kolikoitaKaytetty) return riveja-1;
    }
}

/// <summary>
/// Aritmeettinen summa, kun a_1 = 1, a_n >= a_1 ja
/// perättäisten alkioden erotus on 1.
/// </summary>
/// <param name="n"></param>
/// <returns></returns>
public static int AritmeettinenSumma(int n)
{
    int summa = 0;
    summa = n * (1 + n) / 2;
    return summa;
}

```



Tehtävän 3 pistejakauma.

## Tehtävä 4 (6 p.)

Tee funktio `Voitto`. Funktio ottaa parametrina int-taulukon hinnat, jossa hinnat[i] on tietyn osakkeen hinta i:ntenä päivänä. Esimerkiksi parametrin arvo [7, 1, 5, 3, 6, 4] tarkoittaa, että ensimmäisenä päivänä hinta on ollut 7, toisena päivänä 1, kolmantena 5 ja niin edelleen.

Haluat maksimoida voittonsi valitsemalla yhden päivän osakkeen ostamiseen ja sitten valitsemalla tulevaisuudessa olevan eri päivän kyseisen osakkeen myymiseen. Funktion tulee palauttaa suurin mahdollinen voitto, jonka voit tästä kaupankäynnistä saada. Jos voittoa ei voi saada, tulee palauttaa 0.

**Esimerkki 1:** Syöte: hinnat = [7, 1, 5, 3, 6, 4], paluuarvo: 5. **Selitys:** Osta päivänä 2 (hinta = 1) ja myy päivänä 5 (hinta = 6), jolloin voitto = 6 - 1 = 5. Huomaa, että ostaminen päivänä 2 ja myyminen päivänä 1 ei ole sallittua, koska sinun täytyy ostaa ennen kuin voit myydä.

**Esimerkki 2:** Syöte: hinnat = [7, 6, 4, 3, 1], paluuarvo: 0. **Selitys:** Tässä tapauksessa mitään kauppooja ei tehdä, joten suurin voitto = 0.

### Rajoitteet

- hinnat.Length on vähintään 1.
- hinnat[i] on vähintään 0.
- Jos käytät sisäkkäisiä silmukoita, voit saada enintään 4 pistettä. Täydet pisteet voi saada vain, jos algoritmi toimii ilman sisäkkäisiä silmukoita, ts. aineisto käydään yhden kerran läpi.

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

Alla on max 4 ja max 6 pisteen versiot.

```
/// <summary>
/// Laskee, mikä on suurin mahdollinen voitto, joka voidaan
/// saada ostamalla ja myymällä osake.
/// </summary>
/// <param name="hinnat">Taulukko, jossa hinnat päivittäin</param>
public static int Voitto4Pistetta(int[] hinnat)
{
    int max = 0;
    for (int i = 1; i < hinnat.Length; i++)
    {
        for (int j = i+1; j < hinnat.Length; j++)
        {
            int voittoNyt = hinnat[j] - hinnat[i];
            if (voittoNyt > max) max = voittoNyt;
        }
    }
}
```

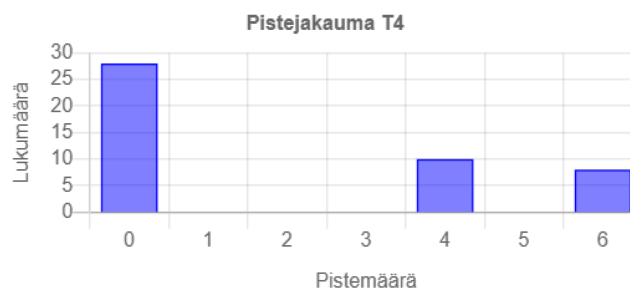
```

    }
}
return max;
}

/// <summary>
/// Laskee, mikä on suurin mahdollinen voitto, joka voidaan
/// saada ostamalla ja myymällä osake.
/// </summary>
/// <param name="hinnat">Taulukko, jossa hinnat päivittäin</param>
public static int Voitto6Pistetta(int[] hinnat)
{
    int voittoMax = 0;
    int minNyt = hinnat[0];

    for (int i = 1; i < hinnat.Length; i++)
    {
        if (hinnat[i] < hinnat[i - 1] && hinnat[i] < minNyt) minNyt = hinnat[i];
        else
        {
            int voittoNyt = hinnat[i] - minNyt;
            if (voittoNyt > voittoMax) voittoMax = voittoNyt;
        }
    }
    return voittoMax;
}
}

```



Tehtävän 4 pistejakauma.

## A Arviointikohteet

### A.1 Tehtävä 3

#	Syöte	Tulos	Virheilmoitus
1	5	2	Viisi kolikkoa, kaksi kokonaista riviä. Kolmannelle riville jää vain kaksi kolikkoa.
2	0	0	Nolla kolikkoa, nolla riviä.
3	1	1	Yksi kolikko, yksi rivi.
4	6	3	Kuusi kolikkoa, kolme kokonaista riviä. Ei jää ylimääräisiä kolikoita.
5	8	3	Kahdeksan kolikkoa, kolme kokonaista riviä. Neljännelle riville jää kaksi kolikkoa.
6	10	4	Kymmenen kolikkoa, neljä kokonaista riviä.

### A.2 Tehtävä 4

#### Syöte-tulos-testit

#	Syöte	Tulos	Pisteet	Virheilmoitus
1	Satunnainen	0	1.0	Funktio ei toimi satunnaisella syötteellä
2	[7, 1, 5, 3, 6, 4]	5	0.5	Suurin voitto on hintojen 1 ja 6 välinen.
3	[7, 6, 4, 3, 1]	0	0.5	Ei ole mahdollista saada voittoa.
4	[7]	0	1.0	Kun syöte on yksittäinen luku, tulos pitäisi olla 0.
5	[7, 7, 6, 5]	0	0.5	Syöte sisältää vain laskevia hintoja, tulos pitäisi olla 0.
6	[7, 7, 5, 6, 7]	2	0.5	Suurin voitto on hintojen 5 ja 7 välinen.

#### Kompleksisuustestit

#	Kompleksisuuskriteeri	Pisteet	Virheilmoitus
1	Ei sisäkkäisiä silmukoita	2.0	Ratkaisussa ei saisi käyttää sisäkkäisiä silmukoita.