

# ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

6. huhtikuuta 2022

## Yleistä

Tentti<sup>1</sup> oli pistekeskiarvon 15,5 (keskihajonta 5,4) perusteella aavistuksen tavanomais-  
ta helpompi. Demohyvitysten kanssa keskiarvo nousi 20,0:een, joten pisterajat nousivat  
tällä kertaa melko korkealle. Huomaa, että demopisteet on laskettu tentin päälle, ja ar-  
vosana lasketaan vasta sen jälkeen. Opiskelijan omat tehtävät ovat nähtävissä TIMissä.  
Uusintojen ajankohdat löydät Sisusta.

Tehtävä	Teki	Keskiarvo	Tarkastaja
T1	147	4,7	Antti-Jussi Lakanen
T2	147	4,2	Antti-Jussi Lakanen
T3	147	3,9	Antti-Jussi Lakanen
T4	136	2,8	Antti-Jussi Lakanen
<b>Yht</b>		<b>15,5</b>	(ennen demopisteitä)

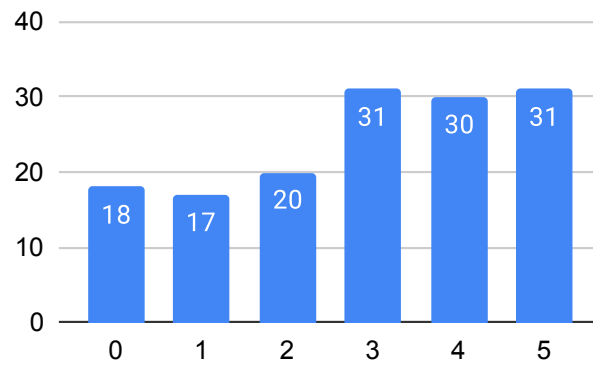
<sup>1</sup><http://users.jyu.fi/~anlakane/ohjelmointi1/tentit/2022-04-06-tentti1.pdf>



JYVÄSKYLÄN YLIOPISTO

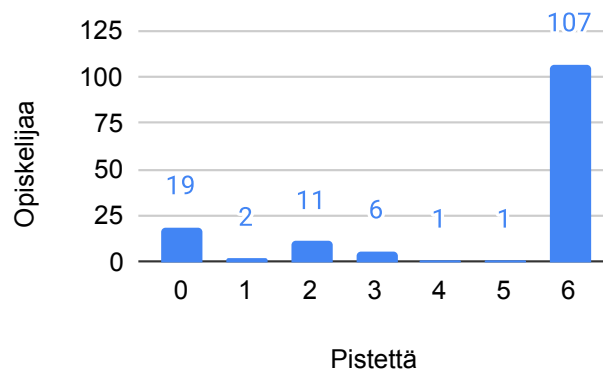
## Arvosteluasteikko ja arvosanajakauma

Arvolause	Pistemäärä (alaraja)
5	26
4	23
3	20
2	16
1	12



Tentin arvosanajakauma.

## Tehtävä 1 (6 p.)



Tehtävän 1 pistejakauma.

### Kysymys 1.1 (3 p.)

Tee funktio `MontakoSuurempaa`. Funktio ottaa parametrina kokonaislukutaulukon, ja palauttaa kuinka moni luvuista on suurempi kuin edellisessä paikassa oleva luku. Esimerkiksi jos taulukko olisi `[3, 4, 1, 1, 2, 5, 4]`, tulos olisi 3, koska luvuista 4, 2 ja 5 kukin on suurempi kuin näitä edeltävässä paikassa oleva luku.

### Malliratkaisu

```
/// <summary>
/// Kuinka moni taulukon luvuista on sellainen,
/// että se on suurempi kuin edellisessä paikassa oleva luku.
/// </summary>
/// <param name="luvut">Luvut.</param>
/// <returns>Suurempien lukujen määrä.</returns>
public static int MontakoSuurempaa(int[] luvut)
{
    int montako = 0;
    for (int i = 1; i < luvut.Length; i++)
    {
        if (luvut[i] > luvut[i - 1]) montako++;
    }
    return montako;
}
```

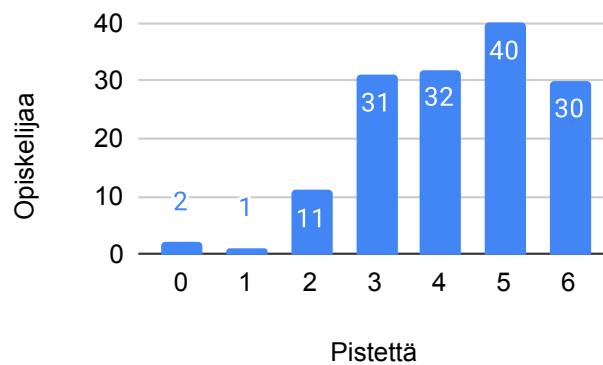
### Kysymys 1.2 (3 p.)

Tee funktio `SuurempienSumma`. Funktio ottaa parametrina kokonaislukutaulukon, ja palauttaa summan luvuista, joista kukin on suurempi kuin edellisessä paikassa oleva luku. Esimerkiksi jos taulukko olisi `[3, 4, 1, 1, 2, 5, 4]` tulos olisi 11, koska  $4 + 2 + 5 = 11$ .

## Malliratkaisu

```
/// <summary>
/// Antaa summan niistä luvuista, joista kukin on suurempi
/// kuin edellisessä paikassa oleva luku.
/// </summary>
/// <param name="luvut">Luvut.</param>
/// <returns>Edellistä suurempien lukujen summa.</returns>
public static int SuurempienSumma(int[] luvut)
{
    int summa = 0;
    for (int i = 1; i < luvut.Length; i++)
    {
        if (luvut[i] > luvut[i - 1])
        {
            summa += luvut[i];
        }
    }
    return summa;
}
```

## Tehtävä 2 (6 p.)



Tehtävän 2 pistejakauma.

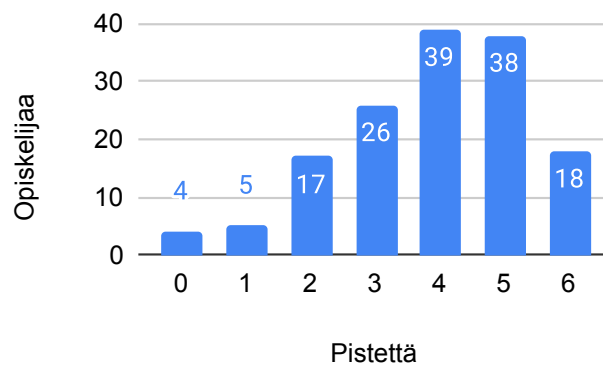
1. Mikä seuraavista väittämistä pitää paikkansa C#-kielen ohjelmointikäytänteiden perusteella?
  - (a) Muuttujan nimessä voi olla puolipiste.
  - (b) Muuttujan nimessä voi olla numeroita.
  - (c) Muuttujan nimi on oltava sama kuin aliohjelman nimi.
  - (d) Muuttujan nimessä toinen ja sitä seuraavat sanat aloitetaan pienaakkosella.
2. Mikä seuraavista väittämistä pitää paikkansa?

- (a) Sijoituslause palauttaa aina bool-tyyppisen arvon.
  - (b) Ohjelmassa on oltava aina vähintään yksi sijoituslause.
  - (c) 'Muuttujan arvon muuttaminen tapahtuu aina sijoitusoperaattorilla '='.
  - (d) Muuttujaan sijoitettava lausekkeen arvo voi koostua vakioista, muuttujien arvoista tai laskutoimituksista.
3. Mikä seuraavista väitteistä pitää paikkansa C#-kielen ohjelmointikäytänteiden perusteella?
- (a) Funktion versiohistoria kirjataan dokumentaatiokommentteihin.
  - (b) Jokaiseen aliohjelman dokumentaatiokommentteihin tulee returns-kenttä.
  - (c) Algoritmin toteutus kuvataan summary-kenttään mahdollisimman yksityiskohtaisesti.
  - (d) Dokumentaatiokommenteissa voi käyttää muitakin kuin ennalta määriteltyjä XML-tageja.
4. Oletetaan, että ohjelma on käynnistetty debuggaustilassa ja sen suoritus on keskeytetty sinun tekemäsi funktion sisälle asetettuun keskeytyskohtaan. Mikä seuraavista väittämistä pitää paikkansa?
- (a) Locals-paneelissa muuttujan arvoksi voidaan sijoittaa funktion kutsu.
  - (b) Muuttujien tyyppejä voidaan muuttaa debuggerin Locals-paneelistä käsin.
  - (c) Main-pääohjelmassa olevien muuttujien arvoja voidaan tarkastella debuggerissa.
  - (d) Watch-paneelissa tehty muuttujan arvon muutos säilyy, kun ajo lopetetaan ja aloitetaan uudestaan.
5. Mikä seuraavista väittämistä pitää paikkansa?
- (a) Taulukon pituutta voi muuttaa taulukon luomisen jälkeen.
  - (b) Taulukon elementit sijaitsevat indekseissä 0...Length.
  - (c) Taulukko voi olla enintään 2-ulotteinen.
  - (d) Taulukon a pituus (a.Length) voidaan päätellä seuraavan lauseen perusteella: `int[] a = new int[] ;`
6. Mikä seuraavista väittämistä pitää paikkansa C#:ssa? Aliohjelman esittelyriville kirjoitetaan...
- (a) parametrien tyypit.
  - (b) lauseke, joka palauttaa paluuarvon.
  - (c) aliohjelman käyttämien attribuuttien luettelo.
  - (d) aliohjelman paikallisten muuttujien tyypit ja nimet.

## Malliratkaisu

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

## Tehtävä 3 (6 p.)



Tehtävän 3 pistejakauma.

1. Mikä seuraavista väittämistä pitää paikkansa?
  - (a) while-silmukan toistoehdon on oltava totuusarvoinen lauseke.
  - (b) while-silmukassa funktiosta voidaan palauttaa (return) useita arvoja.
  - (c) while-silmukka on ainoa silmukkarakenne, jolla voi tehdä ikuisen silmukaan.
  - (d) while-silmukasta voidaan poistua vasta kun kaikki tietorakenteen alkiot on käyty läpi.
2. Mikä seuraavista väittämistä pitää paikkansa? Huomautus: for-silmukan määrittelyrivi koostuu for-sanasta, kaarisulkeista sekä kaarisulkeiden sisään kirjoitettavista lauseista ja lausekkeista. Eräs for-silmukan määrittelyrivi voisi olla:

```
for (int i = 0; i < 4; i++)
```

Toistoehto on silmukan määrittelyrivin keskimmäinen lauseke, yllä esimerkissä  $i < 4$ .

- (a) for-silmukassa tulee aina jossain kohdassa lukea  $i++$ .
  - (b) for-silmukan runko-osaa toistetaan niin kauan kuin toistoehdon arvo on epätosi.
  - (c) for-silmukalla tietorakenteen läpikäynti voidaan aloittaa mistä kohtaa rakennetta tahansa.
  - (d) Mikäli for-silmukan määrittelyrivillä alustetaan muuttuja `int i = 1`, tietorakenteen ensimmäistä alkioita *\*ei\** voida käsitellä.
3. Olkoon meillä oheinen funktion esittelyrivi. Mikä seuraavista väitteistä on pääteltävissä esittelyrivin perusteella?

```
public static double Keskiarvo(List<int> luvut)
```

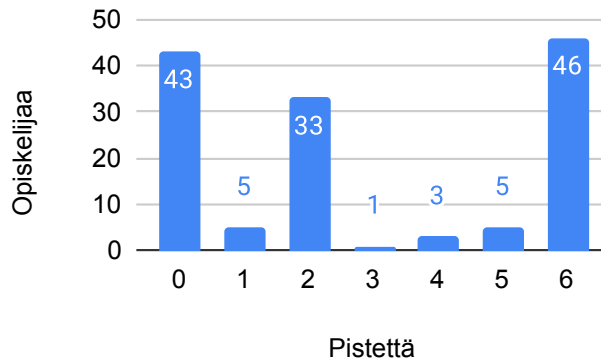
- (a) Funktio voi palauttaa arvon `int.MinValue`.
  - (b) Funktiossa käydään luvut-lista läpi silmukassa.
  - (c) Funktiossa poistetaan osa luvut-listan alkioista.
  - (d) Mikäli luvut-listan sisältö on 0, 3, palauttaa funktio arvon 1.
4. Mikä seuraavista väittämistä pitää paikkansa? Huomautus: Argumentti-sanalla viitataan aliohjelmakutsussa aliohjelmalle annettavaan arvoon, ja parametri-sanalla viitataan aliohjelman määrittelyrivillä määriteltävään muuttujaan.
- (a) On mahdollista, että aliohjelmakutsun argumentteina on pelkkiä aliohjelmakutsuja.
  - (b) Jos aliohjelmassa kutsutaan toista aliohjelmaa, tulee sisemmässä kutsussa antaa vähintään yksi argumentti.
  - (c) Jos aliohjelman paluuarvon tyyppi on jotain muuta kuin `void`, niin sillä tulee olla vähintään yksi parametri.
  - (d) Aliohjelman parametrimuuttujan nimi täytyy olla sama kuin aliohjelmakutsun argumenttina olevan muuttujan nimi.
5. Mikä seuraavista väittämistä pitää paikkansa?
- (a) Funktio voi palauttaa useita paluuarvoja.
  - (b) Aliohjelmassa, jonka tyyppi on `void`, täytyy olla aina tulostuslause.
  - (c) Aliohjelmassa, jonka tyyppi on `void`, voi olla useita `return`-lauseita.
  - (d) Funktion tulee palauttaa saman tyyppinen arvo, kuin mitä jokin sen parametreista on.
6. Mikä seuraavista väittämistä pitää paikkansa?
- (a) `string`-taulukossa voi olla enintään kaksi ulottuvuutta.
  - (b) `string`-taulukon alkioiksi voi sijoittaa myös `StringBuilder`-olion.
  - (c) `string`-taulukon alkioden sisällön muuttaminen aiheuttaa ajonaikaisen virheen.
  - (d) Kaksiulotteinen `string`-taulukko voidaan käydä läpi `foreach`-silmukkaa käyttäen.

## Malliratkaisu

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

## Tehtävä 4 (6 p.)

Tee funktio `TarkistaSulut`. Funktio ottaa parametrina merkkijonon, joka sisältää avaavia ja sulkevia kaarisulkeita, ja palauttaa tiedon, ovatko sulut järjestelty oikein.



Tehtävän 4 pistejakauma.

1. Kullekin avaavalle ja sulkevalle kaarisululle pitää löytyä vastinpari. Esimerkiksi syöte “()” palauttaisi True, sillä sekä avaavia että sulkevia sulkuja on yhtä monta. Sen sijaan syöte “(())” palauttaisi False, koska avaavia sulkuja on kaksi, mutta sulkevia on vain yksi. Vastaavasti “())” palauttaisi False, sillä avaavia sulkuja on vain yksi, mutta sulkevia on kaksi.
2. Sulut tulee avata ja sulkea oikeassa järjestyksessä. Esimerkiksi syöte “((()))” palauttaisi True, sillä kukin sulkeva kaarisulku tulee vasta avaavan parin jälkeen. Sen sijaan syöte “())” palauttaisi False, sillä sulkeva sulku ei voi tulla ennen avaavaa sulkuja.
3. Tyhjä syöte palauttaa True.

Voit olettaa, että syöte sisältää vain kaarisulkeita.

## Malliratkaisu

Malliratkaisun ensimmäisessä versiossa apumuuttuja (int sulkuja) pitää yllä lukua sulkemattomista sulkupareista. Avattu sulku kasvattaa apumuuttujan arvoa yhdellä ja sulkeva sulku pienentää sitä. Tämä muuttuja ei saa mennä alle nollan, sillä muutoin syötteessä olisi sulkeva sulku ennen avaavaa. Toisaalta arvon täytyy funktion lopuksi olla tasan nolla.

```

/// <summary>
/// Tarkistaa, onko aukeavat ja sulkevat sulut oikeassa järjestyksessä.
/// Sulku pitää avata ennen kuin sen voi sulkea. Jokainen avattu sulku
/// pitää sulkea.
/// </summary>
/// <param name="jono">Sulkuja sisältävä merkkijono.</param>
/// <returns>Ovatko sulut syntaktisesti oikein.</returns>
public static bool TarkistaSulut(string jono)
{
    int sulkuja = 0;
    for (int i = 0; i < jono.Length; i++)
    {
        if (jono[i] == '(') sulkuja++;
    }
}

```



```

        else sulkuja--;
        if (sulkuja < 0) return false;
    }
    if (sulkuja != 0) return false;
    return true;
}

```

Toinen vaihtoehto olisi tehdä lista, johon kerätään avaavia sulkumerkkejä. Kun syötteessä kohdataan sulkeva sulku, poistetaan listan viimeinen alkio. Tämä ratkaisu on helpommin laajennettavissa käsittelemään erilaisia sulkumerkkejä.

```

public static bool TarkistaSulut(string jono)
{
    List<char> avatut = new List<char>();
    for (int i = 0; i < jono.Length; i++)
    {
        if (jono[i] == '(')
        {
            avatut.Add(jono[i]);
        }
        else
        {
            if (avatut.Count == 0) return false;
            avatut.RemoveAt(avatut.Count - 1);
        }
    }
    if (avatut.Count > 0) return false;
    return true;
}

```

## Arvostelu

Arvostelu tehtiin automaattisesti seuraavien arvostelukohteiden/testitapausten kautta. Arvostelukohteilla on tärkeysjärjestys: Mikäli ratkaisu ei toteuta ylempanä olevaa kohdetta, ei sitä alempia arvostelukohteita oteta huomioon. Pisteet pyöristettiin alasepäin lähimpään kokonaislukuun.

Syöte	Tulos	Selitys	Virhe	Pistemäärä
""	True	Tyhjä syöte palauttaa True.	Tyhjän syötteen (tyhjä jono) tulisi palauttaa True.	0.5
"()"	True	Yhdet sulut oikeassa järjestyksessä.	Yhdet sulut oikeassa järjestyksessä tulisi palauttaa True.	0.5
"(())"	True	Kahdet sisäkkäiset sulut.	Kahdet sulut sisäkkäin tulisi palauttaa True.	0.5
"((()))"	False	Avaavia sulkuja on enemmän kuin sulkevia.	Jos kaikille avaaville suluille ei löydy sulkevia vastineita, tulisi palauttaa False.	0.5
"((((())))"	True	Useammat sisäkkäiset sulut.	Useat sulut sisäkkäin tulisi palauttaa True.	0.5
"(())"	True	Monimutkaisempi sulkuyhdistelmä.	Sisäkkäisiä sulkuja.	1.0
")"	False	Avaavia sulkuja ei ole, mutta sulkevia sulkuja on.	Jos avaavia sulkuja on vähemmän kuin sulkevia sulkuja, tulisi palauttaa False.	0.5
"))"	False	Avaavia sulkuja ei ole, mutta sulkevia sulkuja on.	Jos avaavia sulkuja on vähemmän kuin sulkevia sulkuja, tulisi palauttaa False.	0.5
"(("	False	Sulkevia sulkuja ei ole, mutta avaavia sulkuja on.	Jos sulkevia sulkuja on vähemmän kuin avaavia sulkuja, tulisi palauttaa False.	0.5
")("	False	Sulut ovat väärässä järjestyksessä.	Jos sulkeva sulku tulee ennen avaavaa sulkua, tulisi palauttaa False.	0.5
"())"	False	Avaavia sulkuja on vähemmän kuin sulkevia sulkuja.	Jos avaavia sulkuja on vähemmän kuin sulkevia sulkuja, tulisi palauttaa False.	0.5