

ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

10. huhtikuuta 2024

Yleistä

Tentti oli pistekeskiarvon 16,6 (keskihajonta 5,3) perusteella tavanomaista helpompi. Demohyvitysten kanssa keskiarvo oli 21,7, joten pisterajat (ks. sivu 2) nousivat poikkeuksellisen korkealle. 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.

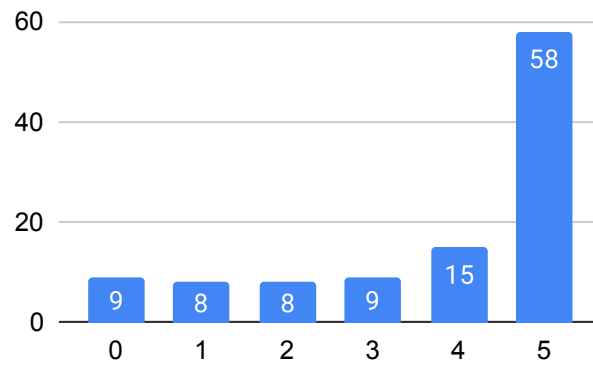
Tehtävä	Teki	Keskiarvo	Automaattinen arvostelu / Tarkastaja
T1	107	3,9	Automaattinen
T2	107	4,9	Automaattinen
T3	107	4,6	Automaattinen
T4	106	3,2	Automaattinen + A-J Lakanen
Yht		16,6	(ennen demopisteitä)



JYVÄSKYLÄN YLIOPISTO

Arvosteluasteikko ja arvosanjakauma

Arvolause	Pistemäärä (inklusiivinen alaraja)
1	12
2	16
3	19
4	21
5	23

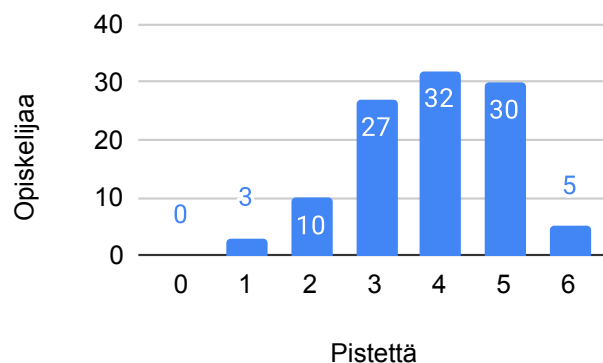


Tentin arvosanjakauma.

Tehtävä 1 (6 p.)

Oikeat vastaukset

- Kysymys 1.1: Funktiolla, jonka paluuarvo on jokin muu kuin void, täytyy olla vähintään yksi return-lause.
- Kysymys 1.2: Taulukon koko voidaan määrittää ohjelman suorituksen aikana.
- Kysymys 1.3: while-silmukan toistoehto tarkistetaan silmukan kierroksen päättyessä continue-lauseeseen.
- Kysymys 1.4: else-osassa oleva koodi suoritetaan vain siinä tilanteessa, että mikään muu ehto ei toteudu.
- Kysymys 1.5: Funktio YhdistaJonot ottaa kaksi parametria, mutta kutsussa on vain yksi argumentti.
- Kysymys 1.6: Funktio laskee virheellisesti summaan vain sellaiset alkiot, jotka eivät ole täsmälleen raja-arvo.



Kuva 1: Tehtävän 1 pistejakauma.

Tehtävä 2 (6 p.)

Arvostelu pohjautui automaattiseen arvioon Liitteessä B Taulukossa 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.

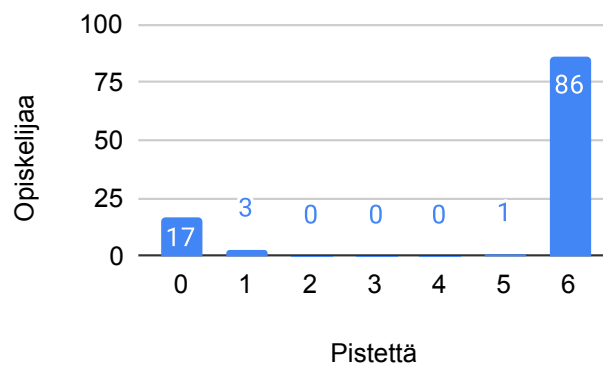
Malliratkaisu

```
/// <summary>  
/// Antaa tentin arvosanan asteikolla 0-5.  
/// Asteikko sisältää arvosanojen 1-5 inklusiiviset alarajat.  
/// Jos pisteet jäävät alle alimman pisterajan, arvosana on 0.  
/// </summary>
```

```

/// <param name="pisteet">Saadut pisteet</param>
/// <param name="asteikko">Arvosanarajat</param>
/// <returns>Arvosana</returns>
public static int LaskeArvosana(int pisteet, int[] asteikko)
{
    if (pisteet < asteikko[0]) return 0;
    for (int i = 1; i < asteikko.Length; i++)
    {
        if (pisteet < asteikko[i]) return i;
    }
    return 5;
}

```



Tehtävän 2 pistejakauma.

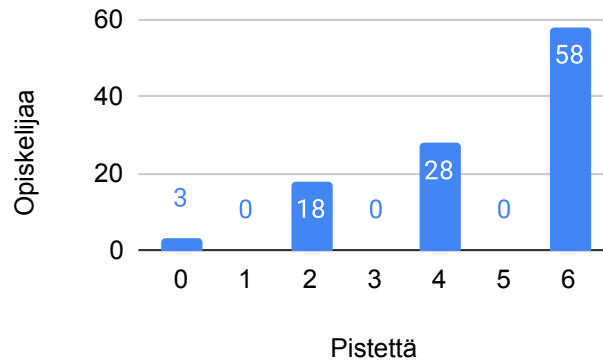
Tehtävä 3 (6 p.)

Oikeat vastaukset

- Kysymys 3.1: Ennen ensimmäistä lukua 99 esiintyvien positiivisten lukujen keskiarvon.
- Kysymys 3.2: 3.
- Kysymys 3.3: Jos taulukko on tyhjä tai lkm-muuttujan arvo on 0.

Tehtävä 4 (6 p.)

Arvostelu pohjautui automaattiseen arvioon Liitteessä B Taulukoissa 2, 3 ja 4 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.

Malliratkaisut

// Tehtävä 4.1

```

/// <summary>
/// Muuttaa jonon tähdiksi.
/// </summary>
/// <param name="jono">Merkkijono</param>
/// <returns>Merkit tähdillä korvattuna</returns>
public static string MuutaTahdiksi(string jono)
{
    // string tahdet = new string('*', jono.Length);
    StringBuilder tahdet = new StringBuilder(new string('*', jono.Length));
    return tahdet.ToString();
}

```

// Tehtävä 4.2

```

/// <summary>
/// Palauttaa uuden merkkijonon, jossa on korvattu tähdillä kaikki
/// sellaiset sanat, jotka ovat täsmälleen samassa muodossa kuin
/// jokin kielletyt-taulukon sanoista.
/// </summary>
/// <param name="teksti">Sensuroitava teksti</param>
/// <param name="kielletyt">Kielletyt merkkijonot</param>
/// <returns>Sensuroitu teksti</returns>
public static string Sensuroi(string teksti, string[] kielletyt)
{
    string[] sanat = teksti.Split(' ');
    for (int i = 0; i < sanat.Length; i++)
    {
        string sana = sanat[i].ToLower();
        foreach (string kielletty in kielletyt)
        {
            if (sana == kielletty)
            {
                sanat[i] = new string('*', sanat[i].Length);
            }
        }
    }
}

```

```

    }
  }
}
return string.Join(" ", sanat);
}

```

Tehtävään 4.3 oli lipsahtanut sikäli virhe, että tehtävässä ei ollut sanottu täytyykö syötteen olla täsmälleen samassa muodossa (suuraakkoset, pienaakkoset) kuin kielletyt sanat. Tästä syystä vastauksiksi hyväksyttiin kummatkin ratkaisut: sellaiset, jotka huomioivat kirjainkoot, ja sellaiset, jotka eivät.

```

// Tehtävä 4.3, versio 1

/// <summary>
/// Mikäli tekstissä esiintyy edes jokin kielletyn sanan alkuosasta
/// (vähintään 4 kielletyn sanan ensimmäistä merkkiä),
/// korvataan tällainen sana tähdillä.
/// Tämä ratkaisu ei huomioi kirjainkokoja, joten esimerkiksi jos
/// aineistossa esiintyy "paha", ja "PAHASANA" on kielletty,
/// niin "paha" korvataan tähdillä.
/// </summary>
/// <param name="teksti">Sensuroitava teksti</param>
/// <param name="kielletyt">Kielletyt merkkijonot</param>
/// <returns>Sensuroitu teksti</returns>
public static string SensuroiParemmiin(string teksti, string[] kielletyt)
{
    string[] sanat = teksti.Split(' ');
    for (int i = 0; i < sanat.Length; i++)
    {
        string sana = sanat[i].ToLower();
        for (int j = 0; j < kielletyt.Length; j++)
        {
            string kielletynAlku = kielletyt[j].ToLower().Substring(0, 3);
            if (sana.StartsWith(kielletynAlku))
            {
                sanat[i] = new string('*', sanat[i].Length);
            }
        }
    }
    return string.Join(" ", sanat);
}

```

```

// Tehtävä 4.3, versio 2

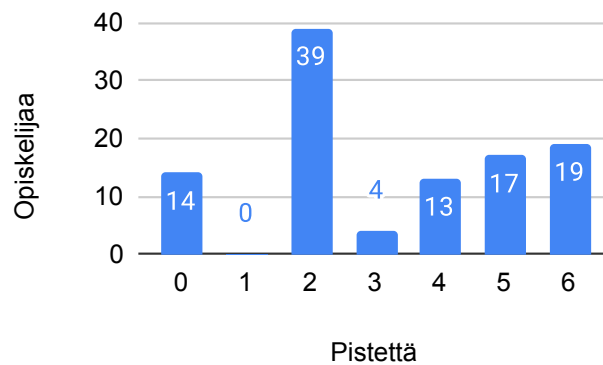
/// <summary>
/// Mikäli tekstissä esiintyy edes jokin kielletyn sanan alkuosasta
/// (vähintään 4 kielletyn sanan ensimmäistä merkkiä), korvataan
/// tällainen sana tähdillä. Tämä ratkaisu huomioi kirjainkoon,
/// joten esimerkiksi jos aineistossa esiintyy "paha", ja "PAHASANA"

```

```

/// on kielletty, niin "paha" EI korvata.
/// </summary>
/// <param name="teksti">Sensuroitava teksti</param>
/// <param name="kielleytyt">Kielletyt merkkijonot</param>
/// <returns>Sensuroitu teksti</returns>
public static string SensuroiParemmiin(string teksti, string[] kielleytyt)
{
    string[] sanat = teksti.Split(' ');
    for (int i = 0; i < sanat.Length; i++)
    {
        foreach (string t in kielleytyt)
        {
            if (sanat[i].StartsWith(t.Substring(0, 4)))
            {
                sanat[i] = new string('*', sanat[i].Length);
            }
        }
    }
    return string.Join(" ", sanat);
}

```



Tehtävän 4 pistejakauma.

A Kysymykset

Ohjelmointi 1 (ITKP102), tentti 10.4.2024 klo 8:00-12:00

(Tästä linkistä pääset takaisin tentin johdantosivulle.)

Kaikki tehtävät koskevat C#-kieltä.

Ohjelmointitehtävissä on luokka ja pääohjelma valmiina, mutta pääohjelma sisältää tarkistuskoodin ja on piilotettu tarkoituksella. Kirjoita palautuslaatikkoon funktio esittelyriveineen ja dokumentaatiokommentteineen tehtävässä esitettyjä ohjeita noudattaen. Halutessasi voit kirjoittaa ComTest-testejä.

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 (6 p.)

Kukin kysymyksistä on yhden pisteen arvoinen. Oikeasta vastauksesta saa yhden pisteen, väärästä vastauksesta nolla pistettä. Kysymyksiin on yksi oikea vastaus.

Kysymys 1.1

Mikä seuraavista väitteistä pitää paikkansa?

- Funktio, joka ei ota parametreja, *ei* voi palauttaa mitään.
- Jos funktio tulostaa jotain, tulostettu asia on samalla myös funktion paluuarvo.
- Funktiolla, jonka paluuarvo on jokin muu kuin `void`, täytyy olla vähintään yksi `return`-lause.
- Funktiolla, jonka paluuarvo on `void` *ei* voi olla useita `return`-lauseita.

Kysymys 1.2

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?

- Taulukon sisältöä *ei* voi muuttaa sen luomisen jälkeen.
- Taulukon koko voidaan määrittää ohjelman suorituksen aikana.
- Lista voi sisältää usean tyyppisiä arvoja, mutta taulukko *ei*.
- Taulukoiden tarkoitus on tallentaa useita eri tyyppisiä arvoja.

Kysymys 1.3

Mikä seuraavista väitteistä pitää paikkansa?

- `for`-silmukassa tulee aina olla vähintään yksi `if`-lause.
- `foreach`-silmukan avulla voidaan käydä läpi taulukon alkiot lopusta alkuun päin.
- `while`-silmukan toistoehto tarkistetaan silmukan kierroksen päättyessä `continue`-lauseeseen.
- `do-while`-silmukkaa käytetään, kun silmukan toistoehto halutaan tarkistaa ennen silmukan suorittamista.

Kysymys 1.4

`if`-lauseen yleinen muoto on seuraava:

```
if (ehto)
{
    // suoritettavaa koodia
}
else if (toinenEhto)
{
    // suoritettavaa koodia
}
else
{
    // suoritettavaa koodia
}
```

Mikä seuraavista väitteistä pitää paikkansa?

- `else`-lause on pakollinen, jos `if`-lauseen ehto ei toteudu.
- Jos ehtolause sisältää `else if`-osan, tulee mukana olla myös `else`-osa.
- `if`-lauseen tarkoitus on suorittaa koodia vain, jos käyttäjä tekee oikean valinnan.
- `else`-osassa oleva koodi suoritetaan vain siinä tilanteessa, että mikään muu ehto ei toteudu.

Kysymys 1.5

Tässä ohjelmassa tulisi yhdistää ja tulostaa kaksi merkkijonoa. Lisäksi ensimmäinen merkkijono tulisi muuttaa isoiksi kirjaimiksi ja toinen pieniksi kirjaimiksi.

Koodiin on lipsahtanut virhe tai virheitä, joka estää ohjelmaa kääntymästä tai toimimasta. Mikä vaihtoehdoista on syy sille, miksi koodi ei toimi?

```
public static void Main()
{
    string sana1 = "kis";
    string sana2 = "sa";
    string yhdistetty = YhdistaJonot(sana1 + sana2);
    Console.WriteLine(yhdistetty); // Pitäisi tulostaa "KISsa"
}

public static string YhdistaJonot(string jono1, string jono2)
{
    return jono1.ToUpper() + jono2.ToLower();
}
```

- Merkkijonoja ei voi yhdistää käyttäen `+` operaattoria.
- Parametrimuuttujien `jono1` ja `jono2` nimet tulisi olla `sana1` ja `sana2`.
- Funktio `YhdistaJonot` ottaa kaksi parametria, mutta kutsussa on vain yksi argumentti.
- `ToUpper` ja `ToLower`-funktioiden paluuarvot tulisi tallentaa muuttujiin ennen merkkijonojen yhdistämistä.

Kysymys 1.6

`RajoitettuSumma`-funktion tulisi laskea yhteen kaikki kokonaislukutaulukon luvut (ensimmäinen parametri) *paitsi* ne alkiot, jotka ovat yhtäsuuria tai suurempia kuin toisena parametrina annettu luku.

Koodiin on kuitenkin lipsahtanut virhe tai virheitä, joka estää ohjelmaa kääntymästä tai toimimasta. Mikä vaihtoehtoista on syy sille, miksi koodi ei toimi?

```
public static void Main()
{
    Console.WriteLine(RajoitettuSumma(new int[] { 1, -2, 3, 9 }, 4));
    // Pitäisi tulostaa 2
}

public static int RajoitettuSumma(int[] taulukko, int raja)
{
    int summa = 0;
    foreach (int t in taulukko)
    {
        if (t != raja) summa += t;
    }
    return summa;
}
```

- Summaa *ei* voi laskea `foreach`-silmukassa.
- `summa`-muuttuja tulee alustaa silmukan sisäpuolella.
- Funktio laskee virheellisesti summaan myös ne alkiot, jotka ovat suurempia kuin raja-arvo.
- Funktio laskee virheellisesti summaan myös ne alkiot, jotka ovat yhtäsuuria kuin raja-arvo.
- Funktio laskee virheellisesti summaan vain sellaiset alkiot, jotka eivät ole täsmälleen raja-arvo.
- Funktiota *ei* voi kutsua siten, että kutsu on `Console.WriteLine`-aliohjelman argumenttina.

Tehtävä 2 (6 p.)

Tee funktio `LaskeArvosana`, joka laskee opiskelijan tenttiarvosanan annetun pistemäärän ja arvosana-asteikon perusteella.

Funktio ottaa kaksi parametria:

- tentin pistemäärä (kokonaisluku)
- tentin arvosana-asteikko (kokonaislukutaulukko). Oletetaan, että arvosana-asteikko sisältää viisi lukua. Luvut ovat arvosanojen alarajojen pistemäärät. Esimerkiksi asteikko [12, 15, 18, 21, 24] tarkoittaa, että arvosana 1 saadaan 12 pisteellä, arvosana 2 saadaan 15 pisteellä jne.

Funktio palauttaa opiskelijan arvosanan asteikolla 0-5.

Esimerkiksi, jos arvosana-asteikko on [12, 15, 18, 21, 24] ja opiskelijan pistemäärä on 20,

funktio palauttaa arvosanan 3.

Tällöin funktiota kutsutaan seuraavasti:

```
int[] asteikko = new int[] { 12, 15, 18, 21, 24 };
int pisteet = 20;
int arvosana = LaskeArvosana(pisteet, asteikko);
Console.WriteLine(arvosana); // Tulostaa 3
```

Tehtävä 3 (6 p.)

Alla on Funktio-niminen aliohjelma. Kysymykset 3.1, 3.2 ja 3.3 liittyvät tähän aliohjelmaan. Kukin kysymyksistä on kahden pisteen arvoinen. Oikeasta vastauksesta saa kaksi pistettä, väärästä vastauksesta nolla pistettä. Kysymyksiin on yksi oikea vastaus.

```
public static double Funktio(int[] luvut)
{
    if (luvut.Length == 0) return 0;
    int lkm = 0;
    int summa = 0;
    foreach (int t in luvut)
    {
        if (t == 99) break;
        if (t <= 0) continue;
        lkm++;
        summa += t;
    }
    if (lkm == 0) return 0;
    return summa * 1.0 / lkm;
}
```

Kysymys 3.1 (2 p.)

Mikä seuraavista kuvaa parhaiten aliohjelman toimintaa?

Aliohjelma palauttaa...

- positiivisten lukujen keskiarvon.
- keskiarvon niistä luvuista, jotka ovat välillä 1-98.
- positiivisten lukujen summan jaettuna taulukon pituudella.
- ennen ensimmäistä lukua 99 esiintyvien positiivisten lukujen keskiarvon.
- ennen ensimmäistä lukua 99 esiintyvien negatiivisten lukujen keskiarvon.
- niiden positiivisten lukujen summan, jotka esiintyvät ennen ensimmäistä lukua 99.
- niiden negatiivisten lukujen summan, jotka esiintyvät ennen ensimmäistä lukua 99.

Kysymys 3.2 (2 p.)

Mikä on aliohjelman paluuarvo, kun sitä kutsutaan seuraavasti?

```
int[] luvut = { 1, 2, 3, 4, 5, 99, 6, 7, 8, 9 };  
double tulos = Funktio(luvut);
```

- 1.5
- 3
- 5
- 19
- Ohjelma kaatuu

Kysymys 3.3 (2 p.)

Missä tilanteessa tai tilanteissa aliohjelma palauttaa arvon 0?

- Jos taulukko on tyhjä tai lkm-muuttujan arvo on 0.
- Jos taulukko on tyhjä, ja lisäksi *joko summa tai lkm* on 0.
- Vain silloin, kun *summa* on 0.
- Vain silloin, kun taulukko on tyhjä.
- Vain silloin, kun taulukossa ei ole lukua 99.
- Vain silloin, kun taulukossa ei ole positiivisia lukuja.

Tehtävä 4 (6 p.)

Työskentelet YouTubella, ja tehtäväksesi on annettu tekstitysten automaattinen sensurointi. Toteutat sensuroinnin vaiheittain.

Kysymys 4.1 (2 p.)

Tee funktio `MuutaTahdiksi`. Funktio ottaa parametrina merkkijonon `jono`, ja palauttaa uuden merkkijonon, jossa kaikki merkit on korvattu tähdillä.

Esimerkki:

```
string jono = "Pahasana";  
string tahdet = MuutaTahdiksi(jono);  
Console.WriteLine(tahdet); // Tulostaa "*****"
```

Voit käyttää seuraavia ComTest-testitapauksia hyödyksesi.

```
MuutaTahdiksi("Pahasana") === "*****";
MuutaTahdiksi("pahasana") === "*****";
MuutaTahdiksi("paha sana") === "*****";
MuutaTahdiksi("")          === "";
MuutaTahdiksi("T")        === "*";
MuutaTahdiksi(" ")        === "*";
```

Vinkki: Uusi merkkijono, jossa on x kpl tähtiä, voidaan luoda esimerkiksi:

```
// x on tähtien määrä - sinun täytyy selvittää mitä x paikalle laitetaan
string tahdet = new String('*', x);
```

```
// tai käyttäen StringBuilderia
StringBuilder sb = new StringBuilder();
sb.Append('*', x);
```

Huomaa, että tässä välilyöntikin muutetaan tähdeksi. Asia korjataan seuraavassa vaiheessa.

Kysymys 4.2 (2 p.)

Tee funktio `Sensuroi`, joka ottaa kaksi parametria:

- merkkijonon `teksti`, joka sisältää sensuroitavan tekstin, sekä
- merkkijonotaulukon `kielleyt`.

Funktio palauttaa uuden merkkijonon, jossa on korvattu tähdillä kaikki sellaiset sanat, jotka ovat täsmälleen samassa muodossa kuin jokin `kielleyt`-taulukon merkkijonoista.

Sovitetaan yksinkertaisuuden vuoksi, että teksti sisältää vain aakkosia a-ö, A-Ö, numeroita sekä välilyöntejä.

Esimerkki:

```
string teksti = "Tämä on tekstiä jossa on pahoja ja TOSI PAHOJA sanoja";
string[] kielleyt = new string[] { "teksti", "pahoja" };
```

```
string sensuroitu = Sensuroi(teksti, kielleyt);
Console.WriteLine(sensuroitu);
// Tulostaa "Tämä on tekstiä jossa on ***** ja TOSI PAHOJA sanoja"
```

Ohje/vinkki:

- Voit käyttää apunasi aiemmin toteuttamaasi `MuutaTahdiksi`-funktioita.
- Voit pilkkoa merkkijonon sanoiksi esimerkiksi `Split`-funktioilla, joka tuottaa taulukon sanoista antamasi erotinmerkin perusteella, esim. `teksti.Split(' ') -> ["Tämä", "on", "tekstiä", "jossa", ...]`.
- Saat yhdistettyä taulukon sanat takaisin yhdeksi merkkijonoksi esimerkiksi `String.Join`-funktioilla, esim. `String.Join(" ", taulukko) -> "Tämä on tekstiä jossa ..."`.

Kysymys 4.3 (2 p.)

Tee funktio `SensuroiParemmiin`. Mikäli tekstissä esiintyy edes jokin kielletyn sanan alkuosasta (vähintään 4 kielletyn sanan ensimmäistä merkkiä), tulee koko sana korvata tähdillä.

Esimerkki:

```
string teksti = "Tämä on tekstiä jossa on pahoja sanoja";
string[] kielletyt = new string[] { "tekst", "paho" };

string sensuroitu = SensuroiParemmiin(teksti, kielletyt);
Console.WriteLine(sensuroitu);
// Tulostaa "Tämä on ***** jossa on ***** sanoja"
```

Ohje/vinkki:

- Voit käyttää esimerkiksi `Substring`-funktioita, jolla voit ottaa merkkijonosta osan, esim. `teksti.Substring(0, 4) -> "Tämä"`.
- Voit käyttää `StartsWith`-funktioita, joka kertoo, alkaako merkkijono toisella merkkijonolla, esim. `teksti.StartsWith("Tämä") -> true`.

Tehtävät päättyvät tähän

(Tästä linkistä pääset takaisin tentin johdantositivulle.)

B Arviointikohteet

Syöte1	Syöte2	Tulos	Selitys	Pistemäärä
12	{ 12, 15, 18, 21, 24 }	1	Pistemäärä antaa arvosanaksi tasan 1	1
12	{ 1, 3, 5, 9, 12 }	5	Arvosana-asteikko voi olla muutakin kuin 12..24	0.5
11	{ 12, 15, 18, 21, 24 }	0	Jos pistemäärä on alle ekan rajan, arvosana on 0	0.5
14	{ 12, 13, 14, 23, 24 }	3	Kaikkien pisterajojen välien ei tarvitse olla saman mittaiset	1
15	{ 12, 15, 18, 21, 24 }	2	Pistemäärä arvosanan 2 arvoinen	0.5
24	{ 12, 15, 18, 21, 24 }	5	Pistemäärä arvosanan 5 arvoinen	0.5
-1	{ 12, 15, 18, 21, 24 }	0	Pisteet voivat olla negatiiviset	1
30	{ 12, 15, 18, 21, 24 }	5	Pisteet voivat olla myös reilusti yli maksimiarvosanan	1

Taulukko 1: Tehtävän 2 arviointikohteet

Syöte	Tulos	Virhe	Pistemäärä
“Pahasana”	“*****”	Kaikki merkit, myös suuraakkoset, korvataan tähdillä	0.5
“pahasana”	“*****”	Kaikki merkit korvataan tähdillä	0.5
“paha sana”	“*****”	Jos jono sisältää useita sanoja, kaikki korvataan	0.5
“”	“”	Tyhjä syöte palautuu tyhjänä	0.25
“T”	“*”	Yksi merkki palautuu yhtenä tähtenä	0.25

Taulukko 2: Tehtävän 4.1 arviointikohteet

Syöte1	Syöte2	Tulos	Selitys	Pistemäärä
“Tämä on tekstiä jossa on pahoja sanoja”	{ “teksti”, “pahoja” }	“Tämä on tekstiä jossa on ***** sanoja”	Kiellettyjen listassa oleva sana tulee sensuroida	0.5
“Tämä on tekstiä jossa on pahoja sanoja”	{ “teksti”, “pahojaa”, “PAHOJA” }	“Tämä on tekstiä jossa on pahoja sanoja”	Vain kiellettyjen listassa, täsmälleen samassa muodossa olevat sanat tulee sensuroida	0.5
“Tämä on tekstiä jossa on paha sana”	{ }	“Tämä on tekstiä jossa on paha sana”	Jos kielletyt on tyhjä, mitään ei muuteta	0.5
“”	{ “teksti”, “pahoja” }	“”	Jos syöte on tyhjä, mitään ei muuteta	0.5

Taulukko 3: Tehtävän 4.2 arviointikohteet

Syöte1	Syöte2	Tulos	Selitys	Pistemäärä
“Tämä on tekstiä jossa on pahoja sanoja”	{ “teksti”, “paho” }	“Tämä on ***** jossa on ***** sanoja”	Sana, jonka alkuosa on kiellettyjen listassa, tulee sensuroida	0.5
“Tämä on tekstiä jossa on pahoja ja TOSI PAHOJA sanoja”	{ “teksti”, “paho” }	“Tämä on ***** jossa on ***** ja TOSI ***** sanoja”	Eri kirjainkoossa olevat sanat muutetaan myös	0.5
“Tämä on tekstiä jossa on paha sana”	{ “teksti”, “pahanlainen” }	“Tämä on ***** jossa on **** sana”	Jos kielletty sana esiintyy vain osittain, se muutetaan myös	0.5
“Tämä on tekstiä jossa on paha sana”	{ }	“Tämä on tekstiä jossa on paha sana”	Jos kiellettyjä sanoja ei ole, teksti ei saa muuttua	0.5

Taulukko 4: Tehtävän 4.3 arviointikohteet