

ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

17. toukokuuta 2019

Yleistä

Tentti¹ oli pistekeskiarvon (14,6) perusteella hieman tavanomaista helpompi. Omasta tehtäväpaperista saa kopion Antti-Jussilta, huone Ag C414.2. Laita sähköpostia tai soita 040 805 3276 etukäteen. Viimeisen uusinnan ajankohdan löydät kurssin Korppi-sivulta.

Tehtävä	Teki	Keskiarvo	Keskihajonta	Tarkastaja
T1	50	4,02	1,52	Mikko Röyskö
T2	50	3,76	1,22	Mikko Röyskö
T3	50	3,68	1,54	Mikko Röyskö
T4	45	3,52	1,87	Mikko Röyskö
Yht		14,6	5,39	

Arvosteluasteikko

Arvolause	Pistemäärä (alaraja)
5	24
4	21
3	18
2	15
1	12

¹<http://users.jyu.fi/~anlakane/ohjelmointi1/tentit/2019-05-17-tentti2.pdf>

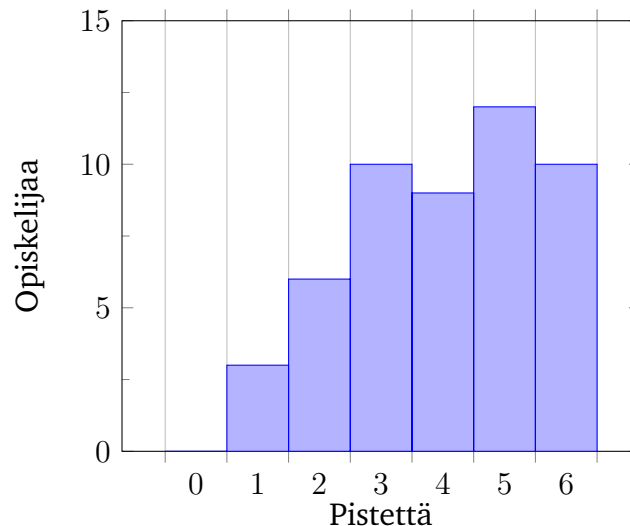


JYVÄSKYLÄN YLIOPISTO

Tehtävä 1 (Tarkastaja: Mikko Röyskö)

Yleiset huomiot

Tehtävä osattiin pistekeskisarvon (4,02) pohjalta melko hyvin. Oikeasta vastauksesta tuli 1 piste, väärästä tai tyhjästä 0 pistettä.



Malliratkaisu

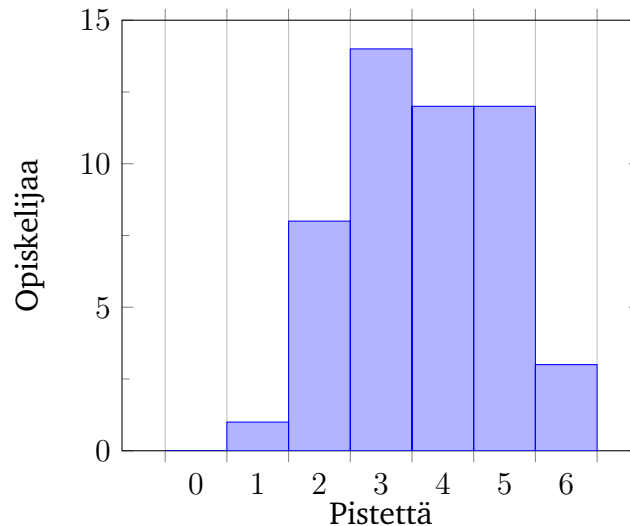
1B, 2A, 3D, 4D, 5D, 6B.

- Kohta 1: Oikea vastaus oli B. Lähes kaikki saivat tämän oikein (n=45, 90%).
- Kohta 2: Oikea vastaus oli A, jonka valitsi 30 tenttijää (60%). D-vaihtoehtoa veikkattiin seuraavaksi eniten (n=13). "Huutomerkkioperaattori" on näistä ainut joka ei muuta käsiteltävänä olevan muuttujan arvoa.
- Kohta 3: Oikea vastaus oli D, jonka valitsi 37 tenttijää (74%). B- ja C-vaihtoehdot molemmat saivat 6 vastausta. Parametrien nimillä ei ole toiminnan kannalta merkitystä.
- Kohta 4: Oikea vastaus oli D, jonka valitsi 29 tenttijää (58%). Seuraavaksi suosituin vastaus oli A (n=14). Tässä piti huomata, että tulostettava luku on aina luvun i toinen potenssi, mutta tätä arvoa ei kuitenkaan tallenneta i -muuttujaan, jolloin i saa arvot 1, 2, 4 ja 8, mutta näiden lukujen toiset potenssit tulostetaan.
- Kohta 5: Oikea vastaus oli D, jonka valitsi 27 tenttijää (54%). Tyypillisesti for-silmukan toistoehdot on epäyhtälö, mutta todellisuudessa siihen voi laittaa minkä tahansa totuuarvon tuottavan lausekkeen, myös funktiokutsun, tai jopa jättää kokonaan pois.
- Kohta 6: Oikea vastaus oli B, jonka valitsi 33 tenttijää (66%).

Tehtävä 2 (Tarkastaja: Mikko Röyskö)

Yleiset huomiot

Oikeasta vastauksesta tuli 1 piste, väärästä tai tyhjästä 0 pistettä.



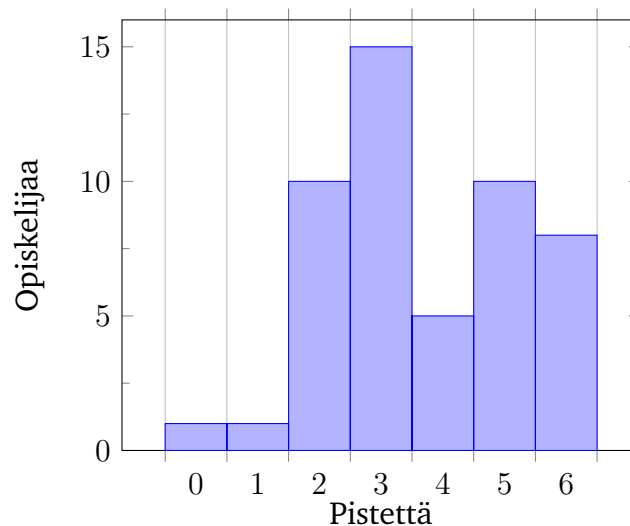
Malliratkaisu

1D, 2C, 3D, 4C, 5C, 6B.

- Kohta 1: Oikea vastaus oli D, jonka vastasi 30 tenttijää (60%). C#-kielessä +-operaattorille voi antaa int-luvun ja string-merkkijonon, ja tuloksena on uusi string-merkkijono, jossa nämä luku ja merkkijono on yhdistetty. Seuraavaksi suosituin vastaus oli C (n=10). Huomaa myös, että joissain ohjelmointikielissä pitää paikkansa että 0=false ja 1=true, mutta ei kuitenkaan C#:ssa.
- Kohta 2: Oikea vastaus oli C, jonka vastasi 44 tenttijää (88%).
- Kohta 3: Oikea vastaus oli D, jonka valitsi 30 opiskelijaa (60%).
 - Lähtötilanne: a=true ja b=false.
 - Käydään D vaihtoehtoa läpi: !a || (a && !b).
 - Sijoitetaan arvot: !true || (true && !false).
 - Lasketaan auki !-operaattorit: false || (true && true)
 - Sievennetään oikea puoli: false || true
 - ||-operaattori palauttaa true jos vähintään toinen arvoista on true.
- Kohta 4: Oikea vastaus oli C, jonka valitsi 31 opiskelijaa (62%). Esittelyrivin pohjalta mikään ei estä muokkaamasta argumenttina annettua listaa. Huomaa kuitenkin, että tämä ei kuitenkaan ole välttämättä suositeltavaa. Jos aliohjelmalle annetaan lista joka sisältää arvot {int.MinValue, int.MinValue}, mitä palautetaan?
- Kohta 5: Oikea vastaus oli C, jonka vastasi 37 opiskelijaa (74%).

- Kohta 6: Oikea vastaus oli B, jonka vastasi 16 opiskelijaa (32%). Tässä oli oleellista huomata että silmukan toistoehdossa oli luvut `.Length`, jonka seurauksena tyhjän taulukon antaminen aliohjelmalle tarkoittaa että silmukkaa ei suoriteta kertaakaan. Täten muuttujan montako arvo on 0, ja ohjelma palauttaa argumentina annetun muuttujan raja arvon.

Tehtävä 3 (Tarkastaja: Mikko Röyskö)



Malliratkaisu

1A, 2B, 3D, 4A, 5A, 6C.

- Kohta 1: Oikea vastaus oli A, jonka valitsi 35 vastaajaa (70%). Aliohjelmakutsun argumenttina voi olla pelkkiä aliohjelmakutsuja, esimerkiksi `Itseisarvo(Keskiarvo(taulukko));`.
- Kohta 2: Oikea vastaus oli B. Tämä osattiin hyvin, 44 oikeaa vastausta (88%).
- Kohta 3: Suosituin vaihtoehto oli B, mutta oikea vastaus oli D, jonka valitsi 17 vastaajaa (34%). Tässä luodaan tyhjä taulukko (vertaa T2.6), jonka pituus on nolla. B-kohdan vastanneet: muistakaa että indeksointi lähtee nolasta, jolloin indeksit menevät `0...Length-1`.
- Kohta 4: Oikea vastaus oli A, jonka vastasi 29 vastaajaa (58%). Esittelyrivillä tulee olla aliohjelman vastaan ottamien parametrien tyypit (ja nimet). Aliohjelman sisällä voi toki luoda uusia (paikallisia) muuttujia ilman niiden kirjoittamista esittelyriville.
- Kohta 5: Oikea vastaus oli A, jonka vastasi 36 vastaajaa (72%). Seuraavaksi suosituin oli C ($n=8$). Aliohjelman paluuarvon voi aina tulostaa, mutta oliotyypeillä järkevä tulostus tarvitsee (monesti) jonkin funktiokutsun avuksi.
- Kohta 6: Oikea vastaus oli C, jonka vastasi 23 vastaajaa (46%). `foreach`-silmukan voi todellakin keskeyttää `break`-lauseella. Seuraavaksi suosituin vastaus oli D jonka vastasi 12 vastaajaa.

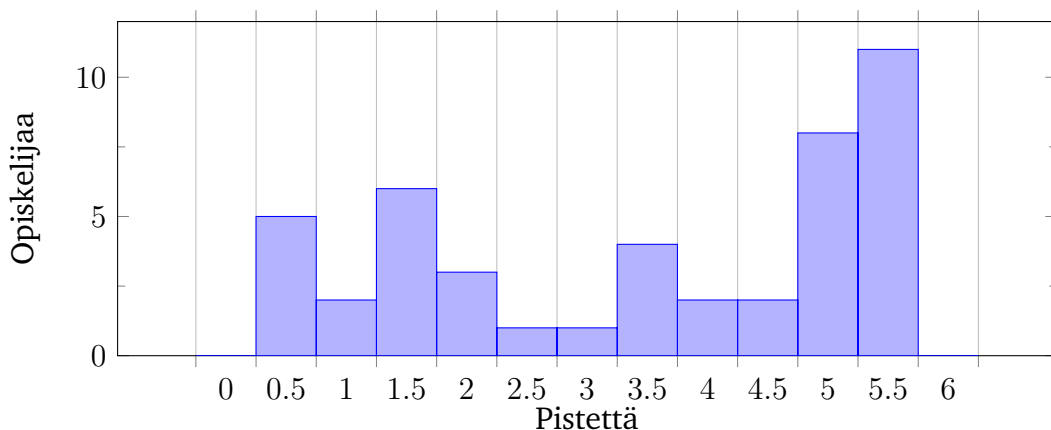
Pisteytys ja virheet

Oikeasta vastauksesta tuli 1 piste, väärästä tai tyhjästä tai usean vaihtoehdon valitsemisesta 0 pistettä.

Tehtävä 4 (Tarkastaja: Mikko Röyskö)

Yleisiä huomioita

- Dokumentaatio osattiin melko hyvin, yleisin virhe oli `<param>`- tai `<returns>`-kohdan puuttuminen.
- Kukaan ei muistanut tarkistaa oliko annettu muuttuja null, jonka seurauksena yksikään vastaaja ei saanut täysiä pisteitä (ks. harjoitus debug-koe).
- Vain muutama ohjelma olisi toiminut suoraan tietokoneelle kopioituna, pienistä syntaksi- ja tyyppivirheistä ei kuitenkaan rangaistu (esim. `taulukko1 == taulukko2`-vertailu ei toimi suoraan halutulla tavalla).
- Unohdettiin muuttujaan sijoitus, esim. oli `sana.toUpperCase()`; eikä `sana = sana.toUpperCase();`.



Tehtävänannossa viiheellisesti puhuttiin anagrammista, mutta selitettiin palindromista jota todellisuudessa pyydyttiin tekemään. Anagrammi on sana, jonka kirjaimet on järjestetty uudelleen (esim. "talouspolitiikka" \longleftrightarrow "tulo pois kaikilta"). Palindromi on taas sellainen sana joka on sama väärinpäin luettuna ("saippuakauppias").

Tehtävänannossa ei määritelty otetaanko kirjainkoko huomioon tai mitä aliohjelman tulee palauttaa, joten monentyylliset vastaukset olivat täysin oikeita.

Malliratkaisu

```
/// <summary>
/// Kertoo onko annettu merkkijono palindromi.
/// Kirjainkoko otetaan huomioon.
/// </summary>
/// <param name="mjono">Merkkijono jota tutkitaan</param>
/// <returns>Oliko annettu merkkijono palindromi</returns>
public static bool Palindromi(string mjono)
```

```

{
    if (mjono == null || mjono.Length <= 1) return false;

    StringBuilder onojm = new StringBuilder("");
    for(int i = 0; i < mjono.Length; i++)
    {
        onojm.Insert(0, mjono[i]);
    }

    if (mjono.Equals(onojm.ToString())) return true;
    else return false;
}

```

Yleisimpiä vaikeuksia

- Silmukkarakennetta käytettiin väärin tai ei ollenkaan.
- Luotiin uusi taulukko johon käytiin lisäämään merkkejä väärinpäin, taulukon koon määrittäminen unohtui, oli väärin tai merkkien lisääminen tehtiin jotenkin väärin.
- Takaperin käännetyt ja alkuperäisen merkkijonon vertailussa oli jokin virhe. Vain ilmiselvistä virheistä rokotettiin.
- Tyhjän/nullin merkkijonon käsittely unohtui.

Pisteytys kohdittain

Keskiarvo/maksimipisteet

d: 0,77/**1,0** p. Kurssin käytännön mukainen XML-dokumentaatio.

f: 0,84/**1,0** p. Funktion esittelyrivi (näkyvyys, paluuarvo, nimi, parametri)

n: 0,27/**1,0** p. Tarkistus tyhjän/nullin merkkijonon varalta

s: 0,68/**1,0** p. Toimiva silmukka

e: 0,59/**1,0** p. Järkevä ehtolause palindromin tarkastuksessa

t: 0,56/**1,0** p. Toimii

v: -0,18/**-1,0** p. Vähennykset

Samasta virheestä ei rangaistu "kahdesti", eli yksi virhe ym. kategorioissa ei vaikuttanut muihin kategorioihin. Esimerkiksi Toimii-kategoriasta pisteen sai mikäli ohjelma "useimmiten" toimii logiikan puolesta. Pistevähennyksiä sai kategorioiden ulkopuolella tehdyistä virheistä (esim. taulukon käyttö väärin) korkeintaan 1 pistettä.

Merkkijonoa yritettiin kääntää takaperin monella eri tavalla. Taulukkoon lisättiin merkki kerrallaan, toiseen string-muuttujaan, tai oikeaoppisesti StringBuilderiin. Ylimääräisten turhien muuttujien käytöstä ei rangaistu.

Monet käyttivät `for`- tai `while`-silmukoita, myös muutama `foreach` näkyi. Muutamalla oli sisäkkäisiä tai perättäisiä silmukoita, nämä ratkaisut kuitenkin useimmiten olivat virheellisiä.

Tehtävässä ei pyydetty kirjoittamaan luokkaa, pääohjelmaa tai testejä, eikä niiden tai minkään ylimääräisen sisällyttämisestä myönnetty tai vähennetty pisteitä.