
Ohjelmointi 1 (kevät 2020), 1. uusintatentti 27.5.2020

Tämän etätentin ohjeet

1. Etätentin suorittaminen

- Etätentti on henkilökohtainen koe. Se suoritetaan yksin ilman ulkopuolisten henkilöiden apua, eikä etätenttiä suoritettaessa saa käyttää suoraan tai välillisesti ulkopuolisten henkilöiden apua.
- Etätenttiin kirjaudutaan yliopiston tunnuksilla. Käyttöehtojen mukaisesti tunnukset ovat henkilökohtaisia, eikä omaa tunnusta saa luovuttaa eteenpäin ulkopuolisille.
- Etätentti suoritetaan sen toteuttamiseen annettuna aikana.

2. Etätentissä sallitut aineistot/materiaalit

- Etätenttiverkkosivulle vastaamiseen saa käyttää tietokonetta, tablettia tai puhelinta.
- Tässä etätentissä saa käyttää tekstiin ja videoon perustuvia verkkosivuja tentin suorittamiseen.
- Tässä etätentissä saa käyttää TIMissä luotua lunttilappua (sähköinen tai tulostettu).
- Tässä etätentissä saa käyttää Visual Studiota, Rideria, tai vastaavaa kehitysympäristöä tentin suorittamiseen.

3. Etätentissä kielletyt välineet/menetelmät

- Koska etätentti on henkilökohtainen koe, etätentin suorittamiseen käytettävää laitetta ei saa käyttää muuhun kommunikointiin.
- Etätenttiä suorittaessa ei saa samanaikaisesti käyttää muita tietojenkäsittely- tai viestintälaitteita.
- Etätentissä käytettävä tietoliikenneyhteys tulee olla tunneloimaton. VPN:n käyttö ei ole sallittua, ja etätentin-selainyhteys tulee ottaa siitä IP-osoitteesta, joka päätelaitteelle on annettu verkkoyhteyden palvelutarjoajalta. IP-osoitetta voidaan käyttää etätenttien suorituksen valvontaan.

- Etätentin aikana verkkoyhteyttä ei saa katkaista. Yhteyden katkaiseminen tai useampien yhteyksien käyttäminen etätentin suorittamiseen voi johtaa suorituksen mitätöimiseen. Mikäli tentin suorittamisen aikana verkkoyhteys on katkennut teknisestä syystä, ilmoita tästä erikseen kurssista vastaavalle opettajalle.

4. Henkilötietojen käsittely etätentin suorittamiseen liittyen

- Yliopisto voi tarkistaa opiskelijan henkilöllisyyden tentin aikana
- Yliopisto voi käyttää etätenttien tenttivalvonnassa ensisijaisesti tietojärjestelmien keräämiä lokitietoja tai live-kuvaa ja toissijaisesti video- ja äänitallenteita.
- Arkaluonteisia henkilötietoja ei käsitellä etätenttien valvonnan yhteydessä.
- Henkilötietojen käsittelyssä noudatetaan yliopiston tietosuoja- ja tietoturvaohjeita.

5. Tenttiin osallistuva opiskelija tiedostaa, että Jyväskylän yliopisto valvoo etätentin suorittamista ja etätenttiohjeiden noudattamista yliopiston käytettävissä olevin keinoin. Opiskelijalla on ehdoton velvollisuus perehtyä ennen tenttiä annettuihin ohjeisiin ja noudattaa annettuja tenttiohjeita yksityiskohtaisesti. Kaikki havaitut poikkeamat annetuista ohjeista tutkitaan ja siirretään vilppiepäilytilanteissa eettisen tutkintaan yliopiston säännösten mukaisesti.

Check your understanding

	True	False
Hyväksyn etätentin ohjeet, ja sitoudun noudattamaan niitä.	<input type="checkbox"/>	<input type="checkbox"/>

Mikäli en hyväksy etätentin ohjeita, ymmärrän, että suoritustani ei arvostella

Muita huomioita

1. Pidä tämä sivu auki koko tentin ajan omassa välilehdessä tai selainikkunassa. Avaa yksittäiset tehtävät (tehtävä 1, tehtävä 2, jne.) avautumisajan jälkeen omiin välilehtiinsä tai ikkunaan (hiiren oikea -> Avaa uuteen välilehteen/ikkunaan).
2. Älä rämpytä refreshiä jos sivun latautuminen kestää, sillä uudelleenlataukset vain hidastavat avautumista. Erityisesti tehtävien sulkeutumis- ja avautumisaikoina Timissä on ruuhkaa joten ole kärsivällinen.
3. **Tallenna vastauksesi ennen palautusajan päättymistä. Mikäli et ole tallentanut vastaustasi ennen palautusajan päättymistä, menetät viimeisimmät muutoksesi.**
4. Tallenna työsi ja päivitä tehtävä sivu n. 5 minuutin välein ihan siltä varalta että tehtävään tulee jokin muutos (esim. bugi tehtävänannossa tai tarkastuskoodissa).
5. Ota kaikki punaiset palkit pois aloittaessasi (ratiaan kuva -> mark all as read), jolloin JOS tulee päivityksiä niin näet keltaisella ne kohdat joita on muutettu.

6. Teknisissä ongelmissa laita ensisijaisesti sähköpostia antti-jussi.lakanen@jyu.fi, toissijaisesti tekstari 040 805 3276 tai soitto.

Tehtävät

Jokaisella tehtävällä on aloitus- ja päättymisaika. Tehtävää ei voi aloittaa ennen sen alkamisaikaa. Tehtävä täytyy tallentaa sille annetussa aikaikkunassa. Sulkeutuneeseen tehtävään ei voi palata.

TALLENNA TEHTÄVÄT ENNEN TEHTÄVÄN PÄÄTTYMISAIKAA.

Voit tallentaa tehtävän useita kertoja. Viimeisin tallennettu vastaus arvioidaan.

Tentissä on neljä tehtävää, joista kaikki tulee tehdä.

Tehtävä 1 (6 p.), alkaa klo 16:00, päättyy 16:20

Tehtävä 2 (6 p.), alkaa klo 16:20, päättyy 16:40

Tehtävä 3 (6 p.), alkaa klo 16:40, päättyy 17:00

Tehtävä 4 (6 p.), alkaa klo 17:00, päättyy 20:00

Tehtävä 1

(Tästä linkistä pääset takaisin tentin pääsivulle.)

Kaikki kysymykset koskevat C#-kieltä.

Kysymys 1.1

Kuinka monta eri arvoa bool-tyyppinen muuttuja voi saada?

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 8



Kysymys 1.2

Oletetaan, että ohjelmassa ei ole `using System;` -riviä. Mikä seuraavista tulostaa tekstin "Hello world"?

- ☐ `Console.WriteLine("Hello world");`
- ☒ `System.Console.WriteLine("Hello world");`
- ☐ `System.Text.Console.WriteLine("Hello world");`
- ☐ `WriteLine` -aliohjelmaa ei voi mitenkään käyttää.



Kysymys 1.3

Mitä silmukkatyyppiä ei voi käyttää taulukon läpikäymiseen, jos silmukan runko-osassa halutaan muuttaa taulukon alkioden arvoja?

- ☐ for
- ☐ foreach
- ☐ while
- ☐ do-while



Kysymys 1.4

Mikä seuraavista väitteistä pitää paikkansa void -tyyppisen (aliohjelma, jonka paluuarvon tyyppi on void) aliohjelmasta?

- ☐ Aliohjelmaa ei voi kuormittaa.
- ☐ Aliohjelmassa ei ole pakko olla return -lausetta.
- ☐ Aliohjelma ei voi ottaa parametreja.
- ☐ Aliohjelmakutsu ei saa olla silmukan sisällä.



Kysymys 1.5

Mikä seuraavista aiheuttaa käännösvirheen?

- ☐ Numero aliohjelman nimen alussa
- ☐ Numero aliohjelman nimen lopussa
- ☐ Skandi (Å/Ä/Ö) aliohjelman nimessä
- ☐ Pienellä alkukirjaimella kirjoitettu aliohjelman nimi



Kysymys 1.6

Mikä seuraavista pitää paikkansa kahdesta sisäkkäisestä for -silmukasta?

- ☐ Kummankin silmukan määrittelyrivillä voidaan alustaa i -niminen muuttuja.
- ☐ Kummassakin silmukassa on käytettävä samannimistä muuttujaa.
- ☐ Sisempää silmukkaa ei välttämättä suoriteta yhtään kierrosta.
- ☐ Ulomman silmukan sisällä ei saa olla return -lausetta.



(Tästä linkistä pääset takaisin tentin pääsivulle.)

Tehtävä 2

(Tästä linkistä pääset takaisin tentin pääsivulle.)

Kysymys 2.1

Mikä seuraavista silmukoista on sellainen, jonka runko-osa suoritetaan ainakin yhden kerran?

- ☒ for
- ☐ foreach
- ☐ while
- ☐ do-while



Kysymys 2.2

Mikä on `a` -muuttujan arvo seuraavan rivin suorittamisen jälkeen?

```
int a = 3 / 4 - 1;
```

- ☒ -1
- ☐ 0
- ☐ 1
- ☐ 0.75



Kysymys 2.3

Mitä `break;` -lauseen jälkeen tapahtuu?

- ☐ Poistutaan silmukasta
- ☐ Jatketaan suoraan silmukan seuraavalle kierrokselle
- ☐ Palautetaan aliohjelman paluuarvo
- ☐ Keskeytetään ohjelman suoritus



Kysymys 2.4

Mikä seuraavista pitää paikkansa, jotta ohjelma kääntyy?

- ☐ `;` -merkillä loppuvat rivit on oltava aina omalla rivillään.
- ☐ `{` -merkin jälkeen (esim. aliohjelmat ja silmukat) pitää tulla aina rivinvaihto.
- ☐ Koko ohjelman voi kirjoittaa vaikka yhdelle riville.
- ☐ Rivinvaihdon saa laittaa ihan mihin vain, vaikka keskelle avainsanaa.



Kysymys 2.5

Mikä seuraavista pitää paikkansa funktion parametrina olevasta `x` -nimisestä muuttujasta?

- ☐ Pääohjelmassa sijaitsevassa funktiokutsussa olevan argumentin on myös oltava nimeltään `x`.
- ☐ Pääohjelmassa sijaitsevassa funktiokutsussa oleva argumentti ei saa olla nimeltään `x`.
- ☐ Muissa aliohjelmissa ja pääohjelmassa voi olla `x` -niminen muuttuja.
- ☐ Muissa aliohjelmissa ei saa olla `x` -nimistä muuttujaa.



Kysymys 2.6

Mihin seuraavista esimerkkitilanteista kannattaisi käyttää % -operaattoria?

- ☐ Muunnetaan tuumat senttimetreiksi.
- ☐ Lasketaan kolmion pinta-ala korkeuden ja kannan perusteella.
- ☐ Lasketaan luokkaretkellä lapsille tasaisesti jaettavien makkaroiden ylijäämä määrä.
- ☐ Lasketaan siitepölyallergisten prosenttiosuus väestössä.



(Tästä linkistä pääset takaisin tentin pääsivulle.)

Tehtävä 3

(Tästä linkistä pääset takaisin tentin pääsivulle.)

Kysymys 3.1

Mikä seuraavista aliohjelmien nimistä noudattaa C#:ssa suositeltua nimeämiskäytäntöä?

- ☐ LaskeKavijamaara
- ☐ laskeKavijamaara
- ☐ laskeKävijämäärä
- ☐ Laske_Kavijamaara
- ☐ laske_Kavijamaara



Kysymys 3.2

Kuinka monta eri arvoa int-tyyppinen muuttuja voi saada?

- ☐ 32
- ☐ 2^{32}
- ☐ 10^{32}
- ☐ noin 2,1 miljardia
- ☐ int -tyypin arvoalue vaihtelee etumerkistä riippuen.



Kysymys 3.3

Jos laitat keskeytyskohdan (*breakpoint*) rekursiivisen aliohjelman toteutuksen ensimmäiselle riville (aaltosulkeen jälkeen ensimmäinen rivi), kuinka monta kertaa ohjelman suoritus pysähtyy, kun aliohjelmää kutsutaan koko ohjelman aikana 10 kertaa?

- ☐ 0
- ☐ 1
- ☐ 9
- ☐ 10
- ☐ 11



Kysymys 3.4

Mikä seuraavista väittämistä pitää paikkansa sellaiselle funktiolle, jonka paluuarvon tyyppi on `string`?

- ☐ Aliohjelman täytyy tulostaa ruudulle `string`-tyyppinen arvo.
- ☐ Aliohjelma voi palauttaa listan `string`-tyyppisiä arvoja.
- ☐ Aliohjelman täytyy ottaa `string`-tyyppinen parametri.
- ☐ Aliohjelma voi palauttaa tyhjän merkkijonon.
- ☐ Paluuarvon sisältö täytyy tulostaa ruudulle pääohjelmassa.



Kysymys 3.5

Kuinka monta kierrosta suoritetaan sellaisen silmukan runko-osa, jonka määrittelyrivi on

```
for (int i = 0; i == 10; i++)
```

kun silmukan sisällä ei ole `break` - tai `return` -lauseita?

- ☐ 0
- ☐ 9
- ☐ 10
- ☐ 11
- ☐ äärettömän monta kierrosta



Kysymys 3.6

Olkoon meillä oheinen funktion esittelyrivi. Mikä seuraavista väitteistä on pääteltävissä esittelyrivin perusteella?

```
public static int Funktio(StringBuilder jono)
```

- ☐ Jos `jono` on tyhjä, funktio ei palauta mitään.
- ☐ Ohjelmassa ei voi olla muita `Funktio` -nimisiä aliohjelmia tai funktioita.
- ☐ `jono` -muuttujan sisällön täytyy muuttua funktion sisällä.
- ☐ Funktio palauttaa jonkin `jono` -muuttujaan liittyvän kokonaislukutyypin ominaisuuden.
- ☐ Funktion kutsuminen pääohjelmasta voi muuttaa pääohjelmassa sijaitsevan `jono` -nimisen `StringBuilder` -muuttujan sisältöä.



(Tästä linkistä pääset takaisin tentin pääsivulle.)

Tehtävä 4

(Tästä linkistä pääset takaisin tentin pääsivulle.)

Alkuluvut ovat sellaisia lukuja, jotka ovat jaollisia vain itsellään ja ykkösellä. Esimerkiksi 5 on alkuluku, koska se on jaollinen vain luvuilla 1 ja 5. Luku 6 taas ei ole alkuluku, koska se on jaollinen myös luvuilla 2 ja 3. On helppo selvittää, onko joku tietty luku alkuluku (ks. tehtävän koodissa `OnkoAlkuluku` -funktio), mutta alkulukujen tuottaminen on vaikeaa.

Panu uskoo löytäneensä funktion, joka tuottaa alkulukuja. Funktio ottaa parametrina kokonaisluvun n , ja palauttaa $n * n + n + 41$, joka on siis väitetyksi aina alkuluku.

Esimerkiksi kokonaisluvulla 4 funktio palauttaa luvun 61 (koska $4 * 4 + 4 + 41 = 61$), joka on alkuluku.

Funktio näyttäisi toimivan ainakin luvuilla 1-10. Panu päättää testata funktion toimivuutta useammalla luvulla, ja kirjoittaa tätä varten seuraavan koodin, jossa luodaan `AlkulukuGeneraattori` -funktiolla (oletettavasti) alkulukuja ja testataan ovatko ne alkulukuja `OnkoAlkuluku` -funktiolla. (Funktioiden toteutukset saat näkyviin *Näytä koko koodi* -linkistä.)

Tutustu Main -pääohjelman toimintaan, ja vastaa koodin jälkeisiin kysymyksiin. Jos muokkaat koodia Timissä, voit palauttaa koodin alkuperäiseksi *Alusta* -linkistä.

```
1 |  /// <summary>
2 |  /// Pääohjelmassa tutkitaan AlkulukuGeneraattori-funktion toimintaa.
3 |  /// Funktio "toimii", jos se palauttaa alkuluvun.
4 |  /// </summary>
5 |  public static void Main()
6 |  {
7 |      int eiToimi = 0;
8 |      const int MAX = 50;
9 |      for (int i = 0; i <= MAX + 1; i++)
10 |     {
11 |         int alkuluku = AlkulukuGeneraattori(i);
12 |         if (!OnkoAlkuluku(alkuluku))
13 |         {
14 |             eiToimi = i;
15 |             break;
16 |         }
17 |     }
18 |
19 |     if (eiToimi == 0) Console.WriteLine("AlkulukuGeneraattori toimii lukuun " + MAX + " asti!");
20 |     else Console.WriteLine("AlkulukuGeneraattori ei toimi luvulla " + eiToimi);
21 | }
```

Kysymys 4.1

Ohjelma tulostaa "AlkulukuGeneraattori ei toimi luvulla 40". Mitä tämä tarkoittaa?

- ☐ AlkulukuGeneraattori -funktio palautti 40, joka ei ole alkuluku. (40 on jaollinen esim. kakkosella.)
- ☐ AlkulukuGeneraattori -funktio ei palauta alkulukua, kun sille annetaan parametrina 40.
- ☐ AlkulukuGeneraattori -funktio palauttaa null, kun sille annetaan parametrina 40.
- ☐ Ohjelma kaatuu, kun AlkulukuGeneraattori -funktiolle annetaan parametrina 40.



Kysymys 4.2

Miksi rivin 11 `if` -lauseessa on `!` -operaattori?

- ☐ Se on huomio koodin lukijalle, että rivi on tärkeä.
- ☐ Se "kääntää", `if` -lauseen ehdon tarkoittamaan `OnkoAlkuluku(alkuluku) == false`
- ☐ Boolean-tyyppisiä arvoja käsiteltäessä `if` -lauseeseen on aina laitettava `!`
- ☐ Funktiota käsiteltäessä `if` -lauseeseen on aina laitettava `!`



Kysymys 4.3

Minkä takia 0 on *huono* alkuarvo `eiToimi` -muuttujalle?

- ☐ 0 ei varmasti ole `alkuluku`.
- ☐ Jos `AlkulukuGeneraattori` -funktiossa jaettaisiin nollalla, ohjelma kaatuisi.
- ☐ Jos `AlkulukuGeneraattori` ei toimisi nollalla, tuloste väittäisi funktion toimivan.
- ☐ Jos `AlkulukuGeneraattori` -funktiossa käytettäisiin jakojäännöstä, tuloste väittäisi virheellisesti, ettei funktio toimi, vaikka se toimisikin.



Kysymys 4.4

Pääohjelman `for` -silmukasta poistetaan `break` -lause. Kuinka monta kierrosta silmukka suorittaa nyt?

- ☐ 49
- ☐ 50
- ☐ 51
- ☐ 52



Kysymys 4.5

Mitä tulosteelle tapahtuu, kun pääohjelman `for` -silmukasta on poistettu `break` -lause?

- ☐ Mikään ei muutu
- ☐ Ohjelma tulostaa nyt kaikki luvut, jotka eivät ole `alkulukuja`
- ☐ Ohjelma tulostaa nyt viimeisimmän luvun, joka ei ollut `alkuluku`
- ☐ Ohjelma tulostaa nyt viimeisimmän luvun, joka tallennettiin `eiToimi` -muuttujaan.



Kysymys 4.6

`MAX` -muuttujasta poistetaan avainsana `const`. Mitä tapahtuu?

- ☐ Mikään ei muutu
- ☐ Ohjelma ei käänny, ellei muuttujan nimeksi vaihdeta `max`
- ☐ Silmukka ei nyt välttämättä huomaa kaikkia tapauksia, joissa `AlkulukuGeneraattori` ei toimi
- ☐ Silmukka suorittaa yhden kierroksen vähemmän



(Tästä linkistä pääset takaisin tentin pääsivulle.)