

81011 Ohjelmointi I – tentti 08.05.2000

Tehtävä 1

Vastaa *ensimmäisenä* tämän tehtävän kysymyksiin:

- (a) Kirjoita nimesi ja opiskelijanumerosi *selkeästi* jokaisen palauttamasi paperin vasempaan yläkulmaan. Jos sinulla ei ole opiskelijanumeroa, kirjoita sen sijaan oppilaitoksesi nimi ja henkilötunnuksesi. [3 p]
- (b) Kopioi seuraava taulukko siististi päällimmäiselle vastauspaperille nimesi ja opiskelijanumerosi alle siten, että jokainen "ruutu" on kooltaan 2×2 konseptipaperin ruutua:

1	2	3	4	5	Σ

 [3 p]

- (c) Mikäli olet *avoimen korkeakoulun opiskelija* ja olet pyrkimässä teekkariksi *hyvän opintomenestyksen avulla*, niin piirrä päällimmäisen vastauspaperisi oikeaan yläkulmaan saman kokoinen \otimes -merkki kuin tämän tehtäväpaperin oikeassa yläkulmassa on. [3 p]

Tehtävä 2

Fibonaccin luvuiksi kutsutaan seuraavan sarjan termejä:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, ...

Eli ensimmäisen ja toisen termin arvo on yksi. Kaikki myöhemmät termit saadaan kahden edeltävän termin summana.

Tee tämän määritelmän pohjalta C++-funktio *fibonacci*, joka laskee ja palauttaa niin monennen Fibonaccin luvun, kuin mitä sen parametri kertoo.

Esimerkiksi

fibonacci(11)

palauttaisi siis arvonaan 89.

Kirjoita funktiosi siististi kaikkien taiteen sääntöjen mukaisesti, sillä jos tarkastajalla on vaikeuksia ymmärtää funktiotasi, se vähentää pisteitä. Esimerkiksi epähavainnolliset muuttujien nimet ovat pahasta. [6 p]

Tehtävä 3

Essee: tiedon etsintä (eli tiedon haku). [6 p]

Tehtävä 4

Selitä lyhyesti (max. 3–5 virkettä) seuraavat käsitteet:

- (a) tietovirta (stream), [1 p]
(b) muuttuja- eli viiteparametri, [1 p]
(c) funktion esittely, [1 p]
(d) staattinen tietorakenne, [1 p]
(e) ohjelman tulkkaminen ja [1 p]
(f) rakenteinen tietotyyppi. [1 p]

Ota vastatessasi huomioon seuraavat:

- Esimerkki ei yksinään riitä vastaukseksi: anna yleinen selitys.
- Älä selitä kysyttyä termillä sen itsensä (tai sen taivutusmuotojen) avulla.
- Selitä yksikäsitteisesti: jos vastauksesi voi tulkita väärin, se tulkitaan väärin.

Tehtävä 5

Oletetaan, että sinulla on prosessoitavana levytiedosto, johon on talletettu kokonaislukuja (yksi luku rivillä). Tiedostossa olevien lukujen määrästä ei ole tarkempaa tietoa. Sama luku voi myös esiintyä tiedostossa useammin kuin kerran. Esitä algoritmi (algoritmikielellä, ei C++:lla), joka selvittää tiedoston *toiseksi suurimman* luvun.

Esimerkiksi, jos tiedostossa olisi luvut:

903, 821, 733, 641, 558, 46, -372, 287, 903 ja -19

antaisi algoritmi tulokseksi 821.

Algoritmin ei tarvitse varautua seuraaviin virhetilanteisiin:

- Tiedostossa on vähemmän kuin kaksi lukua.
- Kaikki tiedostossa olevat luvut ovat samoja. [6 p]