

OHJ-1101 Ohjelmointi Ie, tentti TAI 2. välikoe

30.1.2009, Essi Lahtinen

Jos olet suorittamassa tenttiä, vastaa kaikkiin tehtäviin. Jos olet suorittamassa 2. välikoetta, vastaa vain tehtäviin 2 ja 3.

Kirjoita jokaiseen vastauspaperiisi selkeästi ylläolevat otsikkotiedot, nimesi, opiskelijanumerosi, koulutusohjelmasi, vuosikurssisi, vastauspaperin järjestysnumero ja jättämiesi vastauspapereiden kokonaismäärä. Jos olet korottamassa aikasempaa suoritustasi, kerro, miltä vuodelta suoritukseksi ovat. *Jätä vastauspapereihisi vähintään 3cm ulkoreunamarginaalit ja ensimmäiseen paperiin vähintään 10 cm yläreunamarginaali.* Kirjoita selkeällä käsialalla. Tentin tarkastaja ei ole hieroglyfien erikoisasantuntija. Onnea tenttiin!

Tehtävä 1

1. Oletetaan, että tarvittavat include-direktiivit jne. ovat käytössä ja määriteltynä on muuttuja `int luvut[ KOKO ]`. Selitä yksinkertaisesti suomeksi, minkä tiedon seuraava ohjelmapätkä tulostaa? (2p.)

```
int luku1 = luvut[ 1 ] - luvut[ 0 ];
int luku2 = 0;
for( int i = 2; i < KOKO; ++i ) {
    if( luku1 < luvut[ i ] - luvut[ i-1 ] ) {
        luku1 = luvut[ i ] - luvut[ i-1 ];
        luku2 = i ;
    }
}
cout << luku2 << endl;
```

2. Miksi edellisen kohdan `for`-silmukassa askeltajan `i` arvot alkavat juurikin 2:sta? (1p.) Mikä virhe tapahtuisi, jos askeltaja aloittaisi 0:sta ja koodi olisi muuten täsmälleen samanlainen? (1p.) Miten tämä virhe ilmenisi ohjelmaa ajettaessa? (1p.)
3. Palautustehtävä 2:ssa piti tarkastaa, että `char`-tyyppisiä alkioita sisältävä taulukko salausavain sisältää kaikki kirjaimet välillä a-z. Yksi tapa toteuttaa tarkastaminen oli tehdä silmukkarakenne, joka käy läpi kaikki kirjaimet välillä a-z, ja laittaa em. silmukkarakenteen sisälle tarkastus, että löytyykö vuorossa oleva kirjain salausavain-tilukosta. Tarkastuksen voi kuitenkin tehdä myös toisella, fiksummalla *algoritmil-la*. Kuvaile tämän fiksumman algoritmin toiminta yksinkertaisesti suomeksi. (2p.) Kerro, miksi se on fiksumpi. (1p.)
4. Määriteltynä on `vector< int > luvut1`; ja `vector< int > luvut2`; joista tiedetään, että kumpikaan ei sisällä samaa alkioita kahteen tai useampaan kertaan. Kirjoita C++-ohjelmapätkä, joka tarkastaa, sisältävätkö `vector`it samat alkiot, kun alkoiden järjestystä ei oteta huomioon. (2p.)
5. Onko edelliseen kohtaan kirjoittamasi ohjelma toteutettu niin, että se tekee mahdollisimman vähän turhaa työtä? Jos on, kerro, miten turha työ on vältetty. Jos ei, kerro, miten ohjelman toimintaa voitaisiin tältä osin järkevöittää. (2p.)

6. Kerro lyhyesti suomeksi, mitä muutoksia kohtaan 4 kirjoittamaasi ohjelmaan pitäisi tehdä, jos ei voitaisi olettaa, että *vectoreissa* ei ole samaa alkia useampaan kertaan? (1p.)

Tehtävä 2

1. Mitä hyötyä on luettelotyypin (*enumeration type*) käyttämisestä ohjelmoinnissa? (2p.)
2. Päätät tehdä ohjelman, joka tallentaa tiedot kaikista omistamistasi elokuvista. Jokaisesta elokuvasta pitää perustietojen, kuten esimerkiksi nimen, vuoden ja keston, lisäksi tallentaa myös, missä formaatissa se on (VHS, DVD, MPG, AVI), sekä kaikkien näyttelijöiden nimet ja roolinimet. Määrittele C++-rakenteet, joita käyttäisit tietojen tallentamiseen. (2p.)
3. Kirjoita C++-ohjelmapätkä, joka tulostaa edellisessä kohdassa määritellystä tietorakenteesta kaikkien niiden elokuvien nimet, joissa Johnny Depp esiintyy. (2p.)

Tehtävä 3

1. Etsi jokin joukko lukuja. (0p.)
2. Keskiarvo joukolle lukuja lasketaan siten, että kyseisten lukujen summa jaetaan lukujen lukumäärällä. Kirjoita C++:aa käyttäen:
 - esittely yleiskäyttöiselle funktiolle, joka ottaa kutsujaltaan joukon lukuja, laskee niiden keskiarvon, jos mahdollista, ja ilmoittaa lasketun keskiarvon kutsujalle. (2p.)
 - minimaalinen pääohjelma, joka kutsuu edellä määriteltyä funktiota laskeakseen keskiarvon *jollekin* joukolle lukuja ja tulostaa tuloksen tai virheilmoituksen, mikäli keskiarvoa ei voida laskea. (2p.)
 - määrittely edellisessä kohdassa esitellylle funktiolle. (2p.)
3. *string*-kirjastossa on alimerkkijonon etsimiseen tarkoitettu funktio *substr*, joka ottaa lähtötietoinaan merkkijonon, jonka alimerkkijonoa etsitään, indeksin, josta alimerkkijono alkaa, ja alimerkkijonon pituuden, ja palauttaa ko. alimerkkijonon. *Suunnittele* (=ei tarvitse koodata) kattava testipääohjelma *substr*-funktiolle. Kerro jokaisesta testistä mitä se testaa. (3p.)
4. Suunnittele funktiojako ohjelmalle, joka toimii vähän samaan tapaan kuin graafisen laskimen piirustusominaisuus, eli sille syötetään funktiota, joiden kuvaajat se piirtää koordinaatistoon. Kerro jokaisesta funktiosta sen kuvaava nimi sekä parilla lauseella, mitä se tekee. (2p.)