

OHJ-1110 Laaja ohjelmointi 1. Tenti 2.4.2012.

Kirjoita jokaiseen vastauspaperiin opiskelijanumero ja nimi. Tentissä ei saa olla mukana laskinta eikä muutakaan materiaalia. Vastatessasi kirjoita mahdollisimman selkeällä käsialalla. Tentin on laatinut Hannu-Matti Järvinen.

Tehtävä 1

- Hiihtokilpailun tuloslaskentaohjelmassa tarvitaan tyyppi, johon voidaan tallettaa seuraavat tiedot: nimi, suoritus aika kokonaisina sekunteina ja tieto siitä, onko kilpailijan doping-tulos positiivinen. Määrittele tällainen tyyppi. Käytä määrittelyssä kuvaavia nimiä. (2p)
- Esittele a-kohdassa tehtyä tyyppiä oleva muuttuja ja alusta se. Alustuksen tulee vastata järkevää lähtötilannetta ennen kilpailua. (1p)
- Sijoita b-kohdan muuttujaan mitattu aika (jokin sekuntimäärä, joka vastaa noin tuntia) sekä dopingin tulos (negatiivinen). (1p)
- Jokaisella kilpailijalla on kilpailunumero. Tietoja etsitään ja käsitellään tämän numeron perusteella. Miten tallettaisit kilpailijat ja heidän tietonsa (käytä apuna a-kohdassa tehtyä tyyppiä, mutta saat muuttaa sitä tarvittaessa)? (2p)
- Kirjoita funktio, joka etsii tekemästäsi d-kohdan talletusrakenteesta kilpailun voittajan (pienin aika) ja palauttaa tämän kilpailijan kilpailunumeron. Pidä huolta, että kaikkien kilpailijoiden kohdalla tarkistetaan, että doping-tulos on negatiivinen. Ääritapauksessa voittajaa ei ole. (4p)

Tehtävä 2

Vakiolla KOKO on jokin positiivinen kokonaislukuarvo. Oletetaan, että on olemassa taulukot.

```
int vasen[KOKO];
int oikea[KOKO];
int kaikki[KOKO * 2];
```

- Taulukoissa *vasen* ja *oikea* on lukuja suuruusjärjestyksessä pienimmästä suurimpaan. Tee aliohjelma *yhdista*, joka kokoaa luvut suuruusjärjestyksessä yhteen taulukkoon *kaikki*. Eli jos taulukon *vasen* sisältö on 1, 5, 5, 10 ja taulukon *oikea* sisältö -4, 6, 8, 12, pitää taulukossa *kaikki* olla alkioit -4, 1, 5, 5, 6, 8, 10, 12, tässä järjestyksessä. Taulukot tulee antaa aliohjelmalle parametreina. (5p)
- Muuta a-kohdassa tekemääsi ohjelmaa siten, että taulukkoon *kaikki* luvut tulevatkin järjestyksessä suurimmasta pienimpään. Eli jos taulukon *vasen* sisältö on 1, 5, 5, 10 ja taulukon *oikea* sisältö -4, 6, 8, 12, pitää taulukossa *kaikki* alkioiden järjestys olla seuraava: 12, 10, 8, 6, 5, 5, 1, -4. (1p)
- Funktiolla on kaksi vector-mallista parametria, joissa on joukko kokonaislukuja. Funktiolla tulee palauttaa tieto siitä, onko ensimmäisessä parametrissa vähemmän, yhtä monta vai enemmän positiivisia lukuja kuin toisessa. Ohjelma ei saa muuttaa parametrejaan. Tee ohjelman esittely (toteutusta ei tarvitse tehdä). Kiinnitä huomiota miten funktion arvo palautetaan ja selitä toiminta tarvittaessa. (2p)

Tehtävä 3

- a. Mitä tarkoittaa testauksessa rivikattavuus? Entä ehtokattavuus? (2p)
- b. Seuraava ohjelma kertoo, millainen kolmio on kyseessä, jos sen sivut ovat a, b ja c. Tämä versio olettaa, että parametrit ovat suuruusjärjestyksessä pienimmistä suurimpaan. Tee (mahdollisimman pieni) testiaineisto (riittää kustakin testistä ilmoittaa parametrit a, b ja c), jolla sekä rivi- että ehtokattavuus saadaan 100 prosenttiin. (4p)

```
void millainen_kolmio(int a, int b, int c)
{
    // Tarkistetaan oletus: a <= b <= c
    if (a > b or b > c) {
        cout << "Parametrien tulee olla järjestyksessä pienimmästä suurimpaan" << endl;
        return; // Parametrit väärin, poistutaan aliohjelmasta
    }
    if (a <= 0 or c >= a + b) {
        cout << "Ei ole kolmio" << endl;
        return; // Poistuu aliohjelmasta
    }
    cout << "Kyseessä on ";
    if (a == b or b == c) {
        if (a == b and b == c) {
            cout << "tasasivuinen ";
        } else {
            cout << "tasakylkinen ";
        }
    }
    if (c * c == a * a + b * b) {
        cout << "suorakulmainen ";
    }
    cout << "kolmio" << endl;
}
```

Tehtävä 4

Ohjelmistotalon työpaikkahaastattelussa pyydetään hakijoita tekemään näyteohjelma, joka kertoo parametrina annetusta perinteisestä taulukosta, onko *ainakin yksi* sen alkioista jaollinen neljällä. Taulukon alkiot ovat etumerkittömiä kokonaislukuja ja taulukon lopussa on loppumerkinä nolla. Kerro jokaisesta ohjelmasta (a-d), toimiiko se vai ei ja selitä toimimattomasta ohjelmasta, miksi ohjelma ei toimi tai antamalla esimerkki, jolla ohjelma laskee väärin. Jos ohjelma toimii, vastaa OK tms., jotta vastaus erottuu tyhjästä vastauksesta. Valitse sitten palkattava hakija (a-d), jonka tekemän toteutuksen tulee olla toimiva ja mahdollisimman selkeä. Perustele selkeys. (4p vaihtoehtojen tarkastelusta, 2p valinta ja perustelu)

```
a    bool ainakin_yksi_on_neljalla_jaollinen(unsigned int par[] )
    {
        unsigned int apu = 1;
        for (int i = 0; par[i] != 0; ++i) {
            apu *= par[i];
        }
        return apu % 4 == 0;
    }
```

```
b    bool ainakin_yksi_on_neljalla_jaollinen(unsigned int par[])
    {
        bool apu = false;
        for (int i = 0; par[i] != 0 and not apu; ++i) {
            apu = par[i] % 4 == 0;
        }
        return apu;
    }
```

```
c    bool ainakin_yksi_on_neljalla_jaollinen(unsigned int par[])
    {
        unsigned int apu = 0;
        int i = 0;
        for (; par[i] != 0; ++i) {
            apu += par[i] % 4;
        }
        return apu == 0;
    }
```

```
d    bool ainakin_yksi_on_neljalla_jaollinen(unsigned int par[])
    {
        int i = 0;
        do {
            if (par[i] % 4 == 0) {
                return par[i] != 0;
            }
            ++i;
        } while (true);
        return true;
    }
```

e.