OHJ-1110 Laaja ohjelmointi 1. Tentti 15.12.2010.

Kirjoita jokaiseen vastauspaperiin opiskelijanumero ja nimi. Vastaa tehtävä 1 ja 2 paperiin 1 ja tehtävät 3 ja 4 paperiin 2. Tentissä ei saa olla mukana laskinta eikä muutakaan materiaalia. Vastatessasi kirjoita mahdollisimman selkeällä käsialalla.
Tentin on laatinut Hannu-Matti Järvinen.

Tehtävä 1 (paperi 1)

a. Mitä eroa on tyypillä ja muuttujalla? (1p)

b. Suunnistuskilpailun tuloslaskentaohjelmassa tarvitaan tyyppi, johon voidaan tallettaa seuraavat tiedot: nimi, suoritusaika kokonaisina sekunteina ja tieto siitä, onko kilpailija leimannut oikeat rastit oikeassa järjestyksessä. Määrittele tällainen tyyppi. Käytä määrittelyssä kuvaavia nimiä. (4p).

c. Esittele b-kohdassa tehtyä tyyppiä oleva muuttuja ja alusta se. Alustuksen tulee vastata järkevää lähtötilannetta ennen kilpailua. (1p)

d. Sijoita c-kohdan muuttujaan mitattu aika (jokin sekuntimäärä, joka vastaa noin tuntia) sekä leimauksen onnistumistieto (tällä kertaa onnistui). (1p)

e. Jokaisella kilpailijalla on kilpailunumero. Tietoja etsitään ja käsitellään tämän numeron perusteella. Miten tallettaisit kilpailijat ja heidän tietonsa (käytä apuna b-kohdassa tehtyä tyyppiä, mutta saat muuttaa sitä tarvittaessa)? (2p)

f. Kirjoita funktio, joka etsii tekemästäsi e-kohdan talletusrakenteesta kilpailun voittajan (pienin aika ja rastit löydetty) ja palauttaa tämän kilpailijan kilpailunumeron. (2p)

Tehtävä 2 (paperi 1)

Ohjelmistotalon työpaikkahaastattelussa pyydetään hakijoita tekemään näyteohjelma, joka kertoo parametrina annetusta perinteisestä taulukosta, ovatko kaikki sen alkiot parittomia lukuja. Taulukon alkiot ovat etumerkittömiä kokonaislukuja ja taulukon lopussa on loppumerkkinä nolla. Kerro jokaisesta ohjelmasta (a-d) , toimiiko se vai ei ja tee tarvittava korjaus (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 vaihoehtojen tarkastelusta, 2p valinta ja perustelu)

a

```
bool on_pariton(unsigned int par[] )
{
    int apu = 1;
    for (int i = 0; par.length() > i; ++i) {
        apu *= par[i];
    }
    return apu % 2 != 0;
}
```

b

```
bool on_pariton(unsigned int par[])
{
    bool apu = true;
    for (int i = 0; par[i] != 0 and apu; ++i) {
        apu = par[i] % 2 == 1;
    }
    return apu;
}
```

c

```
bool on_pariton(unsigned int par[])
{
    unsigned int apu = 0;
    int i = 0;
    for (; par[i] != 0; ++i) {
        apu += par[i] % 2;
    }
    return apu == i;
}
```

d

```
bool on_pariton(unsigned int par[])
{
    int i = 0;
    do {
        if (par[i] % 2 == 0) {
            return par[i] == 0;
        }
        ++i;
    } while (true);
    return true;
}
```

Tehtävä 4 (paperi 2)

a. Mitä tarkoittaa testauksessa rivikattavuus? Entä ehtokattavauus? (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. (Pisteitä puoli pistettä tarpeellista testiä kohti. Maksimi sen mukaan.)

```
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;
}
```