

OHJ-1101 Ohjelmointi Ie, tentti

12.3.2007, Essi Lahtinen

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

Tehtävä 1

Selitä *lyhyesti* mitä seuraavat termit tarkoittavat:

- | | |
|---|---|
| 1. literaali (<i>literal</i>) (1p.) | 4. tietue (<i>struct</i>) (1p.) |
| 2. do-while-silmukka (<i>do-while loop</i>) (1p.) | 5. linkkeri (<i>linker</i>) (1p.) |
| 3. break (1p.) | 6. tyyliopas (<i>style guide</i>) (1p.) |

Tehtävä 2

Vastaa *lyhyesti* esimerkiksi ranskalaisia viivoja käyttäen.

1. Mitä eroa on seuraavilla ohjelmapätkillä? (1p.)

| | |
|--|--|
| <pre>// Ohjelmapätkä 1: char c = ' ' cin.get(c);</pre> | <pre>// Ohjelmapätkä 2: char c = ' ' cin >> c;</pre> |
|--|--|

2. Mitä hyötyä on siitä, että kääntäjä antaa virheilmoitusten lisäksi myös varoituksia? (2p.)
3. Koodattuna on funktio `string korvaa(string miono, int alkuindeksi, int pituus, string uusi)`, joka korvaa parametrien alkuindeksi ja pituus määräämän pätkän merkkijonosta `miono` merkkijonolla `uusi`. Ensimmäinen testi ko. funktiolle on suunniteltu seuraavasti:

```
Testi 1: Testataan, että korvaaminen onnistuu yksinkertaisissa tapauksissa. Esimerkkejä testitapauksista:  
korvaa( "abcdefg", 2, 3, "xyz");  
korvaa( "abcdefghijklmn", 4, 6, "xyzääö");  
korvaa( "abcd", 0, 4, "efgh");
```

Suunnittele loput testit *samaan tapaan* ja kerro, mitä niissä testataan. (1p./testi, yht. 3p.)

Kahden pisteen (x_1, y_1) ja (x_2, y_2) kautta kulkevan suoran kulmakerroin saadaan laskettua kaavalla $k = \frac{y_2 - y_1}{x_2 - x_1}$, mikäli $x_1 \neq x_2$, ja suoran yhtälöksi muodostuu tällöin $y - y_1 = k(x - x_1)$. Tässä tehtävässä x_1, y_1, x_2 ja y_2 voivat olla kokonaislukuja. Kirjoita C++:aa käyttäen:

1. esittely yleiskäyttöiselle funktiolle, joka ottaa kutsujaltaan kahden pisteen koordinaatit ja ilmoittaa kutsujalle, missä kohdassa annettu suora leikkaa X-akselin. (2p.)
2. määrittely edellisessä kohdassa esitellylle funktiolle. (2p.)
3. pieni testipääohjelma, joka *vain* kutsuu edellä määriteltyä funktiota laskeakseen yhden suoran ja X-akselin leikkauspisteen ja tulostaa laskennan tuloksen tai virheilmoituksen, mikäli yksikäsitteistä leikkauspistettä ei ole. (2p.)

Tehtävä 4

1. Mitä hyötyä seuraavasta määrittelystä on? `const int KOKO = 10;` (2p.)
2. Oletetaan, että tarvittavat include-direktiivit jne. ovat käytössä ja määriteltynä on muuttuja `int luvut[KOKO]`. Selitä *yksinkertaisesti* suomeksi, mitä seuraava ohjelmapätkä tekee? (2p.)

```
bool tarkastus = true;
for( int i = 0; i < KOKO - 1; ++i ) {
    if( luvut[ i ] < 0 && luvut[ i + 1 ] < 0 ) {
        tarkastus = false;
    }
}
```

3. Mikä virhe seuraavassa ohjelmapätkässä on? Miten korjaat virheen? (3p.)

```
int kohta = 0;
int arvo = 0;
cin >> kohta;

if( kohta < 0 && kohta > KOKO ) {
    cout << "Ei alueella!" << endl;
} else {
    arvo = lukuja[ kohta ];
}
```

4. Kirjoita ohjelmapätkä, joka tarkastaa, onko yksikään taulukossa olevista kokonaisluvuista taulukossa vähintään kahteen kertaan. (3p.)
5. Kohdassa 4 toteutetun toimenpiteen voi toteuttaa useammilla eri algoritmeilla. Kuvaile lyhyesti suomeksi (ei C++-koodina) jokin toinen algoritmi kuin se, minkä toteutit (algoritmin perusidea riittää). Vertaile toteuttamaasi algoritmia ja tätä toista algoritmia. Perustele. (2p.)