

Tehtävä 4

Ajatelaa seuraavanlaisista dynaamisesti varatuista Solu-structeista:

```
struct Solu {
    int data;
    Solu* seuraava;
};
```

koostuvaa linkitettyä listaa, jonka ensimmäiseen alkioon on osoitin muuttujassa:

```
Solu* ensimmainen;
```

Kyseiselle listalle on määritelty algoritmi:

```
Solu* apu1 = ensimmainen;
Solu* apu2 = ensimmainen->seuraava;
Solu* apu3 = apu2;
while ( apu3 != 0 ) {
    apu1->seuraava = apu3->seuraava;
    apu1 = apu1->seuraava;
    apu3->seuraava = apu1->seuraava;
    apu3 = apu3->seuraava;
}
```

- (a) Piirrä laatikko–nuoli-kaaviona tilanne silmukan suorituksen loppuessa, kun alkuperäinen *ensimmainen*-osoittimen määräämä lista sisälsi alusta–loppuun -järjestyksessä luvut 1, 2, 3, 4 ja 5. Huomioi piirroksessa myös se, mihin apuosoittimet osoittavat. [5p]
- (b) Algoritmi on kuitenkin tarkemmin tutkittuna virheellinen. Missä tilanteissa se toimii väärin ja millä tavoin? [2p]
- (c) Mitä muutoksia algoritmiin tulisi tehdä, jotta se toimisi oikein? (Kirjoita koko algoritmi uudelleen ja alleviivaa muuttamasi ja/tai lisäämäsi kohdat). [3p]

Tehtävä 5

Suunnittele *STL-säiliöiden avulla* tietorakenne, johon voitaisiin tallentaa ennalta määräämätön määrä kaupunkien välistä etäisyysinformaatiota. Tietorakenteesta pitäisi pystyä helposti selvittämään, kuinka pitkä matka kaupungista A on kaupunkiin B. Tarkoitus on, että rakenteesta haetaan vain sellaisten kaupunkien välisiä etäisyyksiä, jotka siihen on suoraan talletettu. Siis jos rakenteessa on tiedot etäisyyksistä Helsinki–Tampere ja Tampere–Jyväskylä, eikä mitään muuta, ohjelman *ei tarvitse osata kertoa* etäisyyttä Helsingistä Jyväskylään.

- (a) Esitä määrittely muuttujalle, johon em. kaltaista etäisyysinformaatiota voisi tallentaa. Saat paremmat pisteet, jos pystyt keksimään tehokkaan rakenteen. Ainakin siinä mielessä tehokkaan, että lineaarista hakua tarvitaan hakuoperaatioissa mahdollisimman vähän jos laisinkaan. Kommentoi rakennetta sen verran, että tarkastaja voi yksikäsitteisesti ymmärtää, kuinka sitä on tarkoitus käyttää. [5p]
- (b) Esitä c++-käsky tai -käskyt, joilla määrittelemäsi muuttujaan lisättäisiin etäisyystieto Tampere–Jyväskylä 187 km. [2p]
- (c) Esitä c++-käsky tai -käskyt, joilla edellä tehdyn lisäyksen jälkeen Tampere–Jyväskylä -etäisyys saataisiin muuttujasta selville. [3p]

Tehtävä 6

Essee: Rajapinnat. Mitä ne ovat? Mitä hyötyjä niillä saavutetaan? Mitä mahdollisuuksia c++-tarjoaa rajapintojen toteuttamiseen? [10p]

Tehtävä 7

Mielipiteesi kurssista? Puutteet ja niiden parannusehdotukset. Jos vastaus on hyvä, asia selkeästi esitetty ja mahdollisiin puutteisiin on myös mietitty parannusehdotus, tehtävästä voi saada yhden bonuspisteen. [1p]