

## 81240 OHJELMIEN TODISTAMINEN (3 ov) Tentti 9.4.2001

Kirjoita selvästi nimesi, opiskelijanumerosi ja sähköpostiosoitteesi, ja allekirjoita vastauspaperisi!

**Vihje:** Jos laskelmasi vaativat sivukaupalla paperia, saatat olla hakoteillä.

1. (a) Kirjoita tilapredikaatti, joka väittää, että taulukossa  $B[1..m]$  on vain negatiivisia lukuja.  
(b) Kirjoita tilapredikaatti, joka väittää, että taulukossa  $B[1..m]$  suurin alkio on juuri ennen pienintä alkioita. (Siis esim.  $B[5]$  on suurin ja  $B[6]$  pienin.)  
(c) Selosta sanallisesti tehtävän 3 ohjelman tarkoitus. Jollekin toiselle ohjelmalle vastaus voisi olla "etsii taulukon  $A[1..n]$  pienimmän alkion ja palauttaa sen paikan muuttujassa  $i$ ", "laskee taulukon  $C[1..n]$  luvut yhteen muuttujaan  $x$ " tms.
2. Laske seuraavat heikoimmat esiehdot.  
(a)  $wp(x := x + y; y := x + y, x = x_0 \wedge y = y_0)$   
(b)  $wp(A[A[i]+1] := 5, A[j] = 5)$   
(c)  $wp(\text{if } i = r \text{ then } i := i - 1 \text{ endif}, p \leq i < r \wedge \forall k: p \leq k \leq i: \forall l: i < l \leq r: A[k] \leq A[l])$
3. Todista seuraava ohjelma täysin oikeaksi. Hyvin epämuodollinenkin todistus tuottaa pisteitä, jos siinä on järkeä.  $A$  on määritelty  $A[1..n]$ .  
$$\langle \forall i, j: 1 \leq i < j \leq n: A[i] \leq A[j] \rangle$$
$$ia := 1; iy := n$$
$$\text{while } ia < iy \wedge A[ia] + A[iy] \neq x \text{ do}$$
$$\quad \text{if } A[ia] + A[iy] < x \text{ then } ia := ia + 1 \text{ else } iy := iy - 1 \text{ endif}$$
$$\text{endwhile}$$
$$\langle (ia < iy \wedge A[ia] + A[iy] = x) \vee (ia = iy \wedge \forall i, j: 1 \leq i < j \leq n: A[i] + A[j] \neq x) \rangle$$
4. Tarkastellaan oheista ohjelmaa.  $A$  on määritelty  $A[ala..ylä]$ . Kaikissa "kyllä" / "ei" -tyyppisissä kysymyksissä perustele vastauksesi (mutta älä kovin pitkästi).  
$$\langle ala < ylä \rangle$$
$$ia := ala; iy := ylä; raja := A[ala]$$
$$\text{while } ia < iy \text{ do}$$
$$\quad \text{while } A[iy] > raja \text{ do } iy := iy - 1 \text{ endwhile}$$
$$\quad \text{while } A[ia] < raja \text{ do } ia := ia + 1 \text{ endwhile}$$
$$\quad \text{if } ia < iy \text{ then } apu := A[ia]; A[ia] := A[iy]; A[iy] := apu; ia := ia + 1; iy := iy - 1 \text{ endif}$$
$$\text{endwhile}$$
$$\langle ala \leq iy < ylä \wedge \forall i: ala \leq i \leq iy: \forall j: iy < j \leq ylä: A[i] \leq A[j] \rangle$$
  
(a) Pysähtyykö ulompi **while**-silmukka varmasti, kun se suoritetaan ensimmäisen kerran? Pysähtyykö se varmasti, kun se suoritetaan myöhemmin?  
(b) Kirjoita kaava, joka kertoo taulukon alkioiden suuruuksista suhteessa muuttujan *raja* arvoon, kun suoritus on **if**-lauseen edessä.  
(c1) Voiko olla  $ia > iy$ , kun suoritus on **if**-lauseen edessä? Miten ohjelma jatkaa, jos niin on?  
(c2) Voiko olla  $ia = iy$ , kun suoritus on **if**-lauseen edessä? Miten ohjelma jatkaa, jos niin on?  
(d1) Voiko olla  $ia = iy - 1$ , kun suoritus on **if**-lauseen edessä? Miten ohjelma jatkaa, jos niin on?  
(d2) Voiko olla  $ia = iy - 2$ , kun suoritus on **if**-lauseen edessä? Miten ohjelma jatkaa, jos niin on?  
(e) Voiko olla  $ia < iy - 2$ , kun suoritus on **if**-lauseen edessä? Miten ohjelma jatkaa, jos niin on?  
(f) Toimiiko ohjelma oikein?
5. (a) Mitä ehtoja käsitteen määritelmän on täytettävä, jotta se olisi hyvä? Tässä "käsite" on esimerkiksi "tilakone", "fifo", "asiakastietue" tai "graafi".  
(b) Miten kannattaa menetellä, jos käsitteen rajaaminen tarkalleen halutulla tavalla on vaikeaa?  
(c) Mikä matemaattinen käsite tai lähestymistapa sopii vastaamaan kysymyksiin tyyliä "toteuttaako tämä suunnittelemani tietorakenne prioriteettijonon?" Tässä ei kysytä esim. todistustekniikkaa, jolla vastauksen voi todistaa, vaan merkitystä sanalle "toteuttaa".

loppu