AS-0.101 C/C++ -ohjelmoinnin peruskurssi / Tentti 7.1.2004 / Hannu Laine

Vastaa neljään kysymykseen! Tentin arvosteluasteikko on 0 – 24 pistettä. Kaikkien kysymysten painoarvo on sama (6 pistettä/tehtävä). **Lähtökohtana on, että kysymyksessä 2 käytetään C++-kieltä ja muissa C:tä.** Jos kuitenkin kysymyksessä 2 noppa on toteutettu oikeaoppisesti abstraktina tietotyyppinä C:llä, pisteitä voi ko. tehtävästä saada 4 pistettä.

1.

Huoneessa tehdään lämpötilamittauksia viidessä eri pisteessä viikon aikana eli seitsemänä päivänä kerran päivässä. **Kirjoita ohjelma,** jolle syötetään ensin ensimmäisenä mittauspäivänä mitatut viisi arvoa, sitten toisena mittauspäivänä mitatut viisi arvoa jne. Kun kaikki mitatut arvot on syötetty, ohjelma selvittää, missä huoneen pisteessä viikon keskilämpötila (eri päivinä mitattujen lämpötilojen keskiarvo) oli suurin. Riittää kun ohjelma tulostaa näyttöön mittauspisteen numeron (1, 2, … tai 5).

Huomautus 1. Tehtävä pitää funktioilla jakaa sopivasti osiin ja tietojen välitys on tehtävä parametreillä. Globaaleja muuttujia ei saa käyttää. 6p

Huomautus 2. Syöttötiedoille ei tarvitse tehdä mitään järkevyystarkastuksia.

2.

C:ssä voidaan generoida satunnaislukuja funktiolla rand. Kutsumalla tätä funktiota aina uudelleen saadaan uusi satunnaisluku satunnaislukujen sarjasta. Ennen kyseisen funktion kutsua pitää satunnaislukugeneraattorille antaa siemenluku funktiolla srand (tämä tehdään ohjelmassa kerran). Usein siemenluvuksi annetaan koneen ajanlaskennassa käytettävä funktion time palauttama arvo, jolloin srand funktiota kutsutaan muodossa srand(time(NULL));

Funktio rand voi antaa samoja lukuja uudelleen (samoin kuin 6-sivuinen noppa heitettäessä). Tämän takia se ei sellaisenaan sovi kaikkiin tilanteisiin. Siksi halutaan tehdä helppokäyttöinen satunnaislukugeneraattori luokkana Tnoppa. Tnoppa luokan nopalla voidaan generoida satunnaisia kokonaislukuja halutulta alueelta. Sille voidaan noppaa konstruoitaessa lisäksi kertoa, saako sama luku esiintyä sarjassa uudelleen vai ei. Lisäksi se voidaan välillä resetoida alkutilaan, jonka jälkeen siihen mennessä jo arvotut numerot eivät enää vaikuta resetoinnin jälkeisiin generaattorilta saataviin lukuihin. Resetoinnissa generaattorille annetaan myös uusi siemenluku, mutta lukualue, jolta luvut arvotaan säilyy (tällaista resetointia tarvitaan esim. tilanteessa, kun lähdetään arpomaan itselle uutta seuraavaa veikattavaa lottoriviä).

Luokalla Tnoppa on siis jäsenfunktiot

Tnoppa (konstruktori)
heita
reset

Konstruktorilla asetetaan satunnaislukujen alueen alaraja ja yläraja, tarvittavan satunnaislukusarjan pituus (lukujen maksimimäärä), sekä tieto saako sarjassa esiintyä duplikaatteja vai ei (sama luku enemmän kuin kerran). Jäsenfunktiolla heita saadaan seuraava satunnaisluku sarjasta. Funktio reset resetoi nopan kuten yllä on esitetty.
Kirjoita luokan Tnoppa esittely ja sen kolmen jäsenfunktion toteutukset.

Huomautus 1. Funktion srand prototyyppi on void srand(unsigned int seed) ja funktion rand prototyyppi on int rand(void). Funktio palauttaa satunnaisen kokonaisluvun väliltä 0 – RAND_MAX (joka voi olla esim 32567).

3.

Long int -tyyppisiä muuttujia (4 tavua eli 32 bittiä) käytetään esittämään 8-numeroisia BCD-muodossa olevia lukuja siten, että vähiten merkitsevä numeromerkki on vähemmän merkitsevässä päässä ja eniten merkitsevä numeromerkki on eniten merkitsevässä päässä. **Kirjoita funktio** muunna, joka muuntaa tällaisen parametrina saamansa luvun "normaaliksi" binääriluvuksi, eli kahden komplementtiesitykseksi.

Huomautus. Luvun BCD-esitys (BCD = Binary Coded Decimal) tarkoittaa, että desimaaliluku esitetään binäärimuodossa siten, että jokainen sen numeromerkki (digit) esitetään neljällä bitillä. Siten yhdessä tavussa (8 bittiä) voidaan esittää kaksinumeroinen luku alueelta 0 – 99. Esimerkiksi luku 72 olisi bittimuodossa silloin 01110010 (72H).

Esimerkki funktion toiminnasta: Jos muuttujaan olisi alustettu sisältö muodossa long int a = 0x12345678; (eli heksadesimaalimuodossa), niin funktiolla muunna voitaisiin muuttaa sen sisältö siten, että siellä olisi muunnoksen jälkeen desimaaliluku 12345678 kahden komplementtiesityksenä. Jos muuttuja a tulostettaisiin muunnoksen jälkeen muodossa printf("%ld", a); niin tulostuisi 12345678. Ilman muunnostahan tulostuisi vastaavassa tapauksessa 305419896.

4.

Kaksipäiseksi jonoksi (deque) sanotaan tietosäiliötä, jolla on seuraavat operaatiot:

initialize
addToFront (lisää alkuun)
addToRear (lisää loppuun)
removeFromFront (ottaa alusta)
removeFromRear (ottaa lopusta)

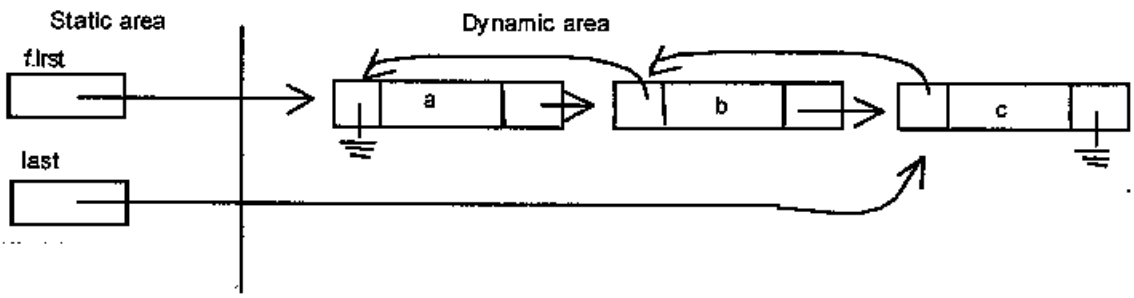
Eräs kätevä toteutustapa tällaiselle kaksipäiselle jonolle on kaksoislinkitetty lista. Alla oleva kuva valaisee asiaa. Kuvassa nähdään tilanne muistissa, kun kaksipäiseen merkkijonoon on viety merkit seuraavilla operaatioilla:

Lisää b alkuun (tai loppuun, koska ensimmäisellä lisäyksellä ei väliä, onko loppuun vai alkuun).
Lisää a alkuun.
Lisää c loppuun.

Toteuta kaksipäinen merkkijono abstraktina tietotyyppinä tällä periaatteella, ts. **kirjoita tietotyypin Tdeque määrittely ja toteuta sen yllämainitut operaatiofunktiot**.

Tietotyyppi Tdeque sisältää siis vain kaksi osoitinta, toinen dynaamisella alueella olevaan ensimmäiseen solmuun (alku) ja toinen viimeiseen (loppu).

Huomautus 1. Erikoistapaukset pitää tietysti hoitaa asianmukaisesti (tyhjä jono, vain yksi alkio jonossa, jne).



5.

Merkkijonot C-kielessä. Selvitä kuinka merkkijonot esitetään C-kielessä ja kuinka niitä käsitellään ohjelmassa (perusperiaatteet ja eroavuudet muihin muuttujatyyppeihin).