

AS-0.100 C/C++ -ohjelmoinnin peruskurssi / Tentti 19.12.2001 / Hannu Laine

Vastaa neljään kysymykseen! Tentin arvosteluasteikko on 0 – 5. Kaikkien kysymysten painoarvo on sama.

Vastaa neljään kysymykseen!!

1.

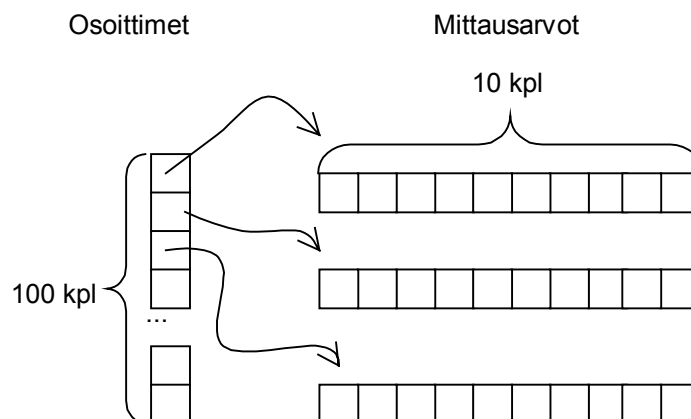
Kirjoita ohjelma, joka toimii seuraavasti: Ohjelmalle syötetään maksimissaan 10 henkilön nimi ja pituus. Kun nimet ja pituudet on syötetty, ohjelma etsii pisimmän henkilön ja tulostaa hänen nimensä ja pituutensa. Tietojen syöttö loppuu, kun syötetään nimeksi “loppu” tai kun tiedot on jo syötetty 10 henkilölle.

Huomautus 1. Tehtävä pitää funktioilla jakaa tehtäväkuvauksen mukaisiin osiin ja tietojen välitys on tehtävä parametreilla. Globaaleja muuttujia ei saa käyttää.

Huomautus 2. Syöttötiedoille ei tarvitse tehdä mitään muuta järkevyytarkastusta kuin se, että tietoja ei syötetä yli 10 henkilölle.

2.

Sovelluksessa käsitellään lämpötilamittauksia. Mittaukset tapahtuvat aina sarjana. Mittaussarjaan kuuluvien mittausten määrää on hyvin vaikea ennustaa etukäteen. Sitä ei tiedetä edes silloin, kun mittaukset kyseiseen sarjaan alkavat. Se kuitenkin tiedetään, että niitä voi olla vähimmillään 10 ja ehdoton maksimi on 1000 mittausta. Mittausten käsittely halutaan tehdä siten, että kovin suuria hukkatiloja ei tulisi varatuksi. Toisaalta ei haluta, että missään vaiheessa jo luettuja mittausarvoja jouduttaisiin siirtämään uuteen paikkaan muistissa. Dynaamisesti linkattuun rakenteeseen ei toisaalta haluta mennä, jotta tilaa ei hukata linkeille. Siksi asia ratkaistaan seuraavalla tavalla. Muistista varataan aluksi tila 100:lle osoittimelle. Itse mittauksille varataan tilaa lisää aina kymmenelle mittaukselle kerrallaan silloin, kun jo varattu tila tulee täyteen. Tällaisten kymmenen mittauksen alueiden osoitteet tallennetaan edellä mainittuun osoitintaulukkoon. Tilanne muistissa näyttäisi alla olevan kuvan mukaiselta, jos mittauksia olisi syötetty 21 – 30.



Kirjoita ohjelma, joka lukee mittausarvoja näppäimistöltä. Mittausarvoiksi kelpaa kaikki positiiviset arvot. Negatiivisen arvon syöttö merkitsee syötön päättämistä. Mittausarvot tulee tallettaa ylläesitetyllä tavalla muistiin. Lopuksi ohjelma tulostaa syötetyt mittausarvot.

Huomautus 1. Ohjelman voi jakaa osiin funktioilla, mutta se ei ole välttämätöntä.

Huomautus 2. Syöttötiedoille ei tarvitse tehdä mitään muita järkevyystarkastuksia kuin, että syötettyjen mittausten lukumäärä ei mene yli 1000. Ohjelman käyttäjälle ei tarvitse tulostaa mitään ohje- tai kehotetekstejä.

3.

Tässäkin tehtävässä käsitellään mittaustapahtumien sarjoja. Yhden mittaustapahtuman tietoihin kuuluu mitattu arvo ja mittaushetken kelloaika.

Tietotyyppi Ttime on määritelty seuraavasti:

```
typedef struct {
    int hours;
    int minutes;
}Ttime;
```

Sinulla on käytettävissä funktio read_time, jolla voidaan lukea tietokoneen kelloaika. Tietotyypin Ttime toinen operaatiofunktio on print_time, jolla kelloaika voidaan tulostaa näyttöön. Funktioiden prototyypit ovat :

```
void read_time(Ttime *time);
void print_time(Ttime time);
```

Yllä oleva tyyppimäärittely ja funktioiden prototyypit ovat otsikkotiedostossa yourtime.h

Määrittele tietotyyppi TOneMeasurement, joka sisältää tiedot yhdestä mittaustapahtumasta (mitattu arvo ja mittauksen kelloaika). **Kirjoita funktio readOneMeasurement**. Tämä funktio lukee näppäimistöltä mittausarvon ja koneelta mittauksen syöttöön liittyvän kelloajan. **Kirjoita vielä funktiot getTime ja getValue**, jotka ovat ns. saantifunktioita. Funktio getTime palauttaa parametrina välitetystä mittaustapahtumatietueesta (tyyppiä TOneMeasurement) mittaustapahtuman kelloajan ja funktio getValue vastaavasti mitatun arvon.

Määrittele tietotyyppi TMeasurementSequence, joka sisältää mittaustapahtumien sarjan. Mittaussarjassa voi olla maksimissaan 100 mittaustapahtumaa. **Kirjoita funktio addMeasurementToSequence**. Se lisää yhden mittaustapahtuman mittaustapahtumien sarjaa. **Kirjoita myös funktio findMaxMeasurement**, jolla on parametrina tietotyyppiä TMeasurementSequence oleva mittaussarja ja joka määrittää sen mittaustapahtuman, jossa mittausarvo on suurin.

Huomautus 1. Tehtävässä on kolme erilaista hierarkian muodostavaa abstraktia tietotyyppiä ja operaatiofunktioiden toteutuksessa on noudatettava ”hyvää ohjelmointitapaa” eli kutakin abstraktia tietotyyppiä käytetään vain sen omilla operaatiofunktioilla.

4.

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.

5.

Vertaile dynaamisia muuttujia ja staattista muuttujia. Vertailussa tarkastellaan ominaisuuksia, käyttöä, mahdollisuuksia, rajoituksia jne. Staattisina muuttujina pidetään kaikkia muuttujia, jotka eivät ole dynaamisessa muistissa (kasassa, heapissä).