

## AS-0.100 C/C++ -ohjelmoinnin peruskurssi / Tentti 8.5.2001 / Hannu Laine

**Vastaa neljään kysymykseen!** Koe arvostellaan skaalalla 0 – 5. kaikkien kysymysten painoarvo arvostelussa on sama.

1.

Meriveden korkeudesta mitataan vuoden aikana kuukausittain korkein arvo ja matalin arvo. Kirjoita ohjelma, jolle syötetään näppäimistöltä ensin mitatut korkeimmat arvot kuukausijärjestyksessä (siis 12 arvoa järjestyksessä tammikuu, helmikuu, jne). Tämän jälkeen ohjelmalle syötetään matalimmat arvot, nekin kuukausijärjestyksessä samalla tavalla. Kun kaikki arvot on syötetty, ohjelma selvittää, missä kuukaudessa korkeimman ja matalimman arvon erotus oli suurin. Riittää, että ohjelma tulostaa näyttöön kuukauden numeron.

Huomautus 1. Ohjelma pitää jakaa funktioilla tehtäväkuvausta vastaaviin osiin ja tietojen välitys on tehtävä parametreilla. Globaaleja muuttujia ei saa käyttää.

Huomautus 2. Syöttötiedoille ei tarvitse tehdä mitään järkevyytarkastuksia.

2.

Kirjoita ohjelma, joka lukee näppäimistöltä merkkijonoja, joiden pituus voi vaihdella 1:sta 50 merkkiin. Aluksi ohjelmalle syötetään näppäimistöltä käsiteltävien merkkijonojen määrä. Tälle määrälle ei ole tiedossa mitään ylärajaa (paitsi että se voidaan esittää tietotyypillä integer). Kun merkkijonojen määrä on syötetty, ohjelmassa aletaan lukea näppäimistöltä itse merkkijonoja. Merkkijonot talletetaan dynaamiselle muistialueelle siten, että kullekin merkkijonolle on varattu juuri sen pituutta vastaava tila. Kun merkkijonot on luettu ja talletettu dynaamiselle muistialueelle, ohjelma tulostaa ne kaikki.

Huomautus 1. Ohjelman toteutuksessa on vältettävä merkkijonojen tallennukseen tarvittavan muistitilan haaskausta.

Huomautus 2. Ohjelman voi jakaa osiin funktioilla, mutta se ei ole välttämätöntä.

Huomautus 3. Syöttötiedoille ei tarvitse tehdä mitään järkevyytarkastuksia.

3.

Tietotyyppi Tcoordinate on määritelty seuraavasti:

```
typedef struct {  
    float x;  
    float y;  
} Tcoordinate;
```

Tälle tietotyypille on määritelty funktiot `read_coordinate`, jolla voidaan lukea paikan koordinaatit näppäimistöltä, funktio `print_coordinate`, jolla voidaan tulostaa paikan koordinaatit näyttöön, ja funktio `distance`, jolla voidaan

laskea kahden koordinaattipisteen välinen etäisyys. Näiden funktioiden prototyypit ovat:

```
void read_coordinate(Tcoordinate *location);  
void print_coordinate(Tcoordinate location);  
void distance(Tcoordinate location1, Tcoordinate location2, float*distance);
```

Ylläesitetyt tietomäärittelyt ja funktioiden prototyypit ovat otsikkotiedostossa `coordin.h`.

Reitti paikasta paikkaan kuljetaan välipaikkakuntien (milestone) kautta. Yhteen välipaikkakuntaan kuuluvat tiedot ovat välipaikkakunnan nimi ja sen suorakulmaiset koordinaatit (metreinä). Yksi reitti muodostuu useasta peräkkäisestä välipaikkakunnasta.

**Määrittele tietotyyppi `TMilestone`**, joka sisältää yhden välipaikkakunnan tiedot. Yhden välipaikkakunnan tietoja ovat sen nimi ja sen koordinaatit. **Kirjoita funktio `readMilestone`**, joka lukee ensin välipaikkakunnan nimen (esimerkiksi Helsinki) näppäimistöltä ja sitten sen sijainnin (koordinaatit). **Kirjoita myös funktio `milestoneDistance`**, joka laskee etäisyyden kahden välipaikkakunnan välillä.

**Määrittele tietotyyppi `TRoute`**, joka sisältää kaikki tietyn reitin kuvaamiseen tarvittavat tiedot (reitin kaikkien peräkkäisten välipaikkakuntien tiedot). Reitillä olevien välipaikkakuntien maksimimäärä tunnetaan ja se on 50 (eri reitit voivat tietysti sisältää eri määrän välietappeja). **Kirjoita funktio `readRoute`**, joka lukee näppäimistöltä yhden reitin tiedot. **Kirjoita myös funktio `calculate_route_lenght`**, jolla on parametrina tietotyyppiä `TRoute` oleva reitti ja joka määrittää reitin pituuden ja palauttaa sen.

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.

Huomautus 2. Kaikissa tapauksissa kahden peräkkäisen pisteen tai paikkakunnan välinen matka tulkitaan suoraksi viivaksi.

Huomautus 3. Tietotyypin `Tcoordinate` operaatiofunktioita tehtävässä ei tarvitse kirjoittaa. Niitä vain käytetään annettujen prototyyppien edellyttämällä tavalla.

4.

Kirjoita funktio, joka saa parametrina merkkitaulukon ja kokonaisluvun, joka ilmoittaa montako merkkiä taulukon alusta käsitellään. Funktio asettaa taulukon merkkeihin pariteettibitin parillisen pariteetin mukaan.

Merkit ovat 7 bittisen standardi ASCII-koodiston mukaisia. Tämä merkitsee, että merkkikoodin bitit ovat bittipositioissa 0, 1, 2, ... 6. Bittipositiossa 7 on

alunperin täysin satunnaista sisältöä, eli siinä voi olla 0 tai 1 täysin muista biteistä riippumatta. Funktio asettaa bittipositioon 7 pariteetin aina parillisen pariteetin mukaan. Tämä merkitsee, että jokaisessa taulukon merkissä olisi funktion suorituksen jälkeen parillinen määrä bittejä, kun lasketaan kaikki kahdeksan bittipositiota

5.

Vertaile funktioita ja pametrillisia makroja.