

AS-0.101 C/C++ -ohjelmoinnin peruskurssi / Tentti 09.05.2005 / Aki Hiisilä

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 merkkikassi on toteutettu oikeaoppisesti abstraktina tietotyyppinä C:llä, pisteitä voi ko. tehtävästä saada 4 pistettä.

Palauta jokainen tehtävä erillisellä konseptilla!

Tehtävä 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 pienin. Riittää kun ohjelma tulostaa näyttöön mittauspisteen numeron (1, 2, 3, 4 tai 5).

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

Huomautus 2. Syöttötiedoille ei tarvitse tehdä mitään järkevyystarkastuksia.

Tehtävä 2

Joukko on tietojen säiliö, jossa yksi tieto voi esiintyä vain kerran. Kassi taas on tietojen kokoelma, johon voidaan tallentaa sama tieto useampia kertoja ja kassi on tietoinen myös kunkin sinne tallennetun tiedon esiintymiskerrasta. Kassi voidaan toteuttaa luokkana (tai abstraktina tietotyyppinä). Tässä tehtävässä toteutetaan merkkien kassi (luokka CharBag), jonne voidaan siis tallentaa yksittäisiä merkkejä. Merkkikassilla on seuraavat jäsenfunktiot

```
CharBag (konstruktori)
add
remove
getCount
```

Konstruktorilla tehdään kassi käyttövalmiiksi ja sen sisältö tyhjäksi. Jäsenfunktiolla *add* voidaan lisätä yksi parametrinä annettu merkki kassiin. Jäsenfunktiolla *remove* voidaan poistaa parametrinä annetun merkin yksi esiintymä kassista ja jäsenfunktiolla *getCount* voidaan kysyä parametrinä annetun merkin esiintymiskertojen lukumäärä kassissa.

Kirjoita luokan CharBag esittely ja sen jäsenfunktioiden toteutukset.

Tehtävä 3

Mitä alla oleva ohjelma tulostaa, kun oletetaan, että yksi tavu on kahdeksan bittiä?

```
#include <stdlib.h>
#include <stdio.h>

void print_1(unsigned char byte);
unsigned char print_2(unsigned char byte);
void modify(unsigned char byte);

int main(void)
{
    unsigned char byte = 0x16;
    unsigned char f;

    printf("A: ");
    print_1(byte);
    printf("\nB: ");
    print_1(byte & 0x8A);
    printf("\nC: ");
    print_1(byte || 0x11);
    printf("\nD: ");
    print_1(byte + 2);
    printf("\nE: ");
    f = print_2(byte);
    printf("\nF: %d\n", f);
    return EXIT_SUCCESS;
}

void print_1(unsigned char byte)
{
    unsigned char mask = 0x80;

    while(mask) {
        if(byte & mask)
            printf("1");
        else
            printf("0");
        mask = mask >> 1;
    }
}

unsigned char print_2(unsigned char byte)
{
    static unsigned char i = 0;
    if(byte) {
        i += print_2(byte >> 1);
        printf("%d", (byte & 0x01) ? 1 : 0);
        return i;
    }
    i = 0;
    return 1;
}
```

Tehtävä 4

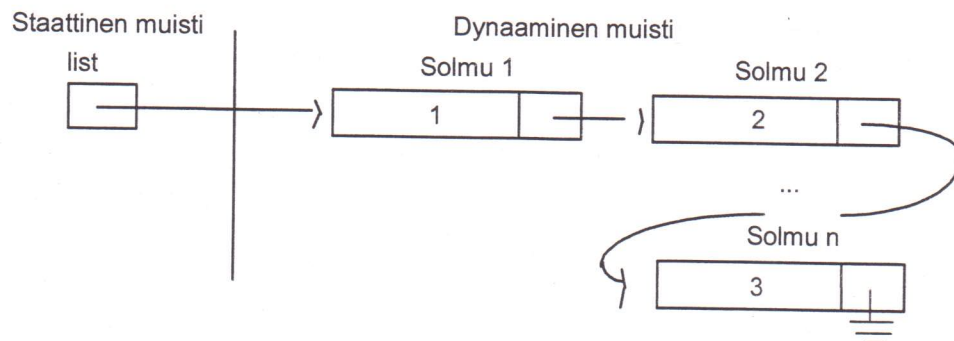
Kokonaislukujen listalle tehdään dynaamisesti linkattu toteutus siten, että koko listaa edustaa yksi osoitin ensimmäiseen dynaamisella muistialueella olevaan solmuun (katso kuva alla). **Kirjoita tarvittavat tietomäärittelyt** tätä listaa edustavalle tyypille TList. **Kirjoita sille lisäksi seuraavat kolme operaatiofunktiota:**

listInitialize, joka alustaa listaa edustavan osoittimen siten, että se sisältää NULL-osoittimen, joka edustaa tyhjää listaa.

listRemove, joka poistaa kokonaisluvun listasta parametrina annetun indeksin kohdalta.

listReverse, joka kääntää parametrina annetun listan järjestyksen päinvastaiseksi. Kääntö pitää tehdä siten, että vain osoittimia modifioidaan.. Listan solmut (nodet) pitää pysyä paikoillaan muistissa eikä solmun datasisältöä (kokonaislukuja) saa siirtää solmusta toiseen.

Huomautus: Erikoistapaukset pitää tietysti hoitaa asianmukaisesti (tyhjä lista, vain yksi alkio listassa jne.). NULL-osoitin edustaa tyhjää listaa.



Tehtävä 5

Alla on toteutettu C-kielellä abstrakti tietotyyppi (ADT) TPerson. Toteutuksessa on joitakin virheitä. Etsi ja perustele virhekohdat, sekä kerro miten korjaisit ne.

```
#include <stdlib.h>
#include <string.h>

typedef struct {
    char *name;
    unsigned int age;
} TPerson;

int personConstruct(TPerson *person)
{
    person = (TPerson *)malloc(sizeof(char *) +
                                sizeof(unsigned int));
    if(person == NULL) {
        return 0;
    }
    return 1;
}

void personSetAge(TPerson *person, unsigned int age)
{
    person->age = age;
}

unsigned int personGetAge(const TPerson *person)
{
    return person->age;
}

int personSetName(TPerson *person, const char *name)
{
    if(name == NULL) {
        person->name = NULL;
        return 1;
    }
    person->name = malloc(strlen(name + 1) * sizeof(char));

    if(person->name == NULL) {
        return 0;
    }
    strcpy(person->name, name);
    return 1;
}

char *personGetName(const TPerson *person)
{
    return person->name;
}

void personDestruct(TPerson *person)
{
    free(person->name);
    free(person);
}
```