

Tentamen Programmeermethoden

Donderdag 10 maart 2016, 14:00–17:00 uur

Universiteit Leiden — Informatica



Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de heading of lokaal voorkomen; vul zelf headings goed in. De te behalen punten (totaal 100) staan tussen haakjes bij de opgaven. Succes! Cijfers: www.liacs.leidenuniv.nl/~kosterstwa/pm/cijf/res.html.

1. (25) In een array `int A[n]` staan n (een `const` ≥ 1) gehele getallen.
 - a. (4) Schrijf een Booleaanse C++-functie `gesorteerd (A,n)` die `true` oplevert precies als `A` oplopend gesorteerd is.
 - b. (5) Schrijf een C++-functie `int hoe (A,n)` die bepaalt hoeveel getallen in array `A` de som van hun twee directe voorgangers zijn. Voor `7 4 5 9 14 4` is dat `2` (namelijk `9` en `14`).
 - c. (6) Schrijf een C++-functie `bouw (A,n,X,Y)` die array `A` als volgt met getallen vult. Vooraan komen de getallen `X` en `Y`. Vervolgens moeten de array-element zo worden dat `hoe (A,n)` een zo groot mogelijke waarde (welke is dat?) gaat geven.
 - d. (6) Schrijf een C++-functie `busort (A,n,k)` die het array `A` met de volgende variant van *bubblesort* oplopend sorteert. De enige aanpassing is dat steeds *na* elke `k`-de ronde met de functie van `a` wordt gekeken of we al klaar zijn, waarna gestopt wordt als dat zo is. (Dus als `k 3` is, wordt *na* `3e`, `6e`, `9e`, ... ronde gekeken.) Neem aan dat $1 \leq k < n$.
 - e. (4) Hoeveel vergelijkingen tussen array-elementen worden minimaal/maximaal uitgevoerd in de functie van `d`? Tel de vergelijkingen in de aanroepen van `a` mee. Geef voor beide situaties een voorbeeld.

2. (25) a. (6) Bij een functie kun je te maken hebben met *call by value* en *call by reference*, en ook met *locale* en *globale* variabelen. Verder onderscheiden we ook nog *formele* en *actuele* parameters. Leg deze zes begrippen duidelijk uit.

b. (6) Gegeven een C++-programma met daarin de volgende twee functies:

```
int ennio (int x, int y, int z) {
    int hulp = x; x = y; y = z; z = hulp;
    cout << x << "," << y << "," << z << endl;
    return x+y+z;
}//ennio
int sergio (int u, int v, int w) {
    int i = 17, som = 0;
    for ( i = 1; i <= w; i++ ) { v--; som += ennio (u,v,w); }//for
    cout << u << "," << v << "," << w << endl; return som;
}//sergio
```

Verder zijn de globale variabelen `x`, `y` en `z` gegeven (alle van type `int`). Voordat de functie `sergio` wordt aangeroepen hebben zij de waarde 5, 6 en 3 respectievelijk. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
cout << sergio (x,y,z) << endl; cout << x << "," << y << "," << z << endl;
```

- c. (3) Druk de return-waarde van `sergio (u,v,3)` uit in `u` en `v`.
- d. (4) Als `b`, maar nu staat er een `&` bij alle zes parameters.
- e. (3) Als `d` (dus weer met zes `&`'s), maar nu met `cout << sergio (x,x,x) << endl;`.
- f. (3) Mag je binnen `ennio` de functie `sergio` aanroepen met `sergio (42,42,hulp);`?

3. (25) Gegeven is een m bij n (beide $\text{const} > 0$; ze hoeven bij deze opgave niet doorgegeven te worden als parameter) array L met gehele getallen ≥ 0 . Hierbij geeft $L[i][j]$ de hoogte ter plaatse (i, j) aan (met $0 \leq i < m$ en $0 \leq j < n$). Een 0 staat voor water. Zie hiernaast voor een voorbeeld waarin $m = 4$ en $n = 5$.

2	0	3	4	0
7	0	4	0	2
1	0	0	9	2
4	5	3	1	1

a. (7) Schrijf een Booleaanse C++-functie `land(L, rij)` die `true` oplevert als er een rij is die geheel uit land bestaat (geen nullen heeft), en de index van die rij in `rij` oplevert. Als er meerdere rijen voldoen, geef degene met de hoogste rij-index. In het voorbeeld: `true` met `rij` gelijk aan 3.

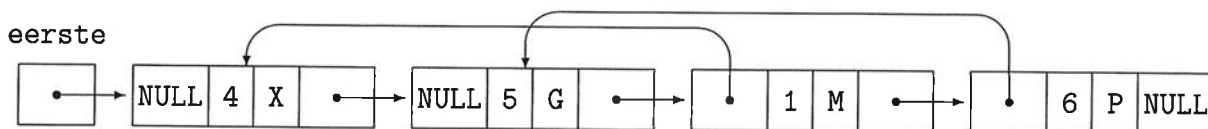
b. (8) Een punt heet een *top* als het niet aan de rand ligt, en een waarde groter dan of gelijk zijn vier directe horizontale en verticale burens heeft. Schrijf een C++-functie `int toppen(L, hoog)` die het aantal toppen in L retourneert, en de waarde van de hoogste in `hoog` oplevert. In het voorbeeld: 2 toppen, en `hoog` wordt 9.

c. (10) We willen de oppervlakte van de zee bepalen die het gegeven punt (i, j) bevat; neem aan dat $L[i][j]$ gelijk is aan 0 (met $0 \leq i < m$ en $0 \leq j < n$). Bij zee gaat het om een zo groot mogelijk gebied met nullen die horizontaal of verticaal grenzen. In het voorbeeld zou voor $(0, 1)$ het antwoord 4 zijn. Schrijf een niet-recursieve functie `int zee(L, i, j)` die dit berekent, bijvoorbeeld door herhaald array-elementen met waarde 0 door -1 te vervangen (en na afloop weer op 0 te zetten).

4. (25) Gegeven is het volgende type:

```
class mens { public: mens* vvorig; int nr; char naam; mens* volg; };
```

Hiermee wordt een lijst van mensen gemaakt (`naam` is een hoofdletter). Het veld `volg` bevat een pointer naar het volgende `mens`-object, en `vvorig` naar het voor-vorige, of `NULL` als dat er niet is. Een voorbeeld (eerste van type `mens*`):



a. (5) Schrijf een C++-functie `voegtoe(eerste, mensnr, mensnm)` die een nieuw `mens`-object met `mensnr` en `mensnm` erin vooraan in de lijst met ingang `eerste` toevoegt. Zet de `vvorig`-pointer van het oude tweede object (als dat bestond) ook goed.

b. (5) Schrijf een C++-functie `verwijder(eerste)` die het *tweede* `mens`-object uit de structuur die door `eerste` wordt aangewezen, netjes verwijdert — mits het bestaat. Zet eventuele `vvorig`-pointers goed.

c. (5) Schrijf een C++-functie `wissel(eerste)` die de twee voorste `mens`-objecten omwisselt, mits ze alfabetisch verkeerd staan (zoals in het voorbeeld). Controleer of de lijst minstens twee objecten heeft. Eventuele `vvorig`-pointers hoeven niet te worden aangepast.

d. (4) In de functies bij **a**, **b** en **c** staat in de heading een pointer. Deze heb je call by value of call by reference doorgegeven (met een `&`). Maakt het voor de werking van deze functies verschil uit of die `&` erbij staat? Mag het, moet het? Leg duidelijk uit.

e. (6) Schrijf een C++-functie `mens* achteraan(eerste, t)` die een pointer oplevert naar het t -de `mens`-object van achteren — mits dat bestaat. Neem aan dat t minstens 1, en hoogstens de lengte van de lijst is. Voor $t = 2$ zou in het voorbeeld een pointer naar het object met `M` erin moeten worden geretourneerd. Loop eerst helemaal naar het einde!