

Tentamen Programmeermethoden

Dinsdag 4 januari 2011, 10.00–13.00 uur

Universiteit Leiden — Informatica

Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de heading of als locale variabele voorkomen; vul zelf headings goed in. De opgaven tellen alle vier even zwaar mee. Succes! Cijfers: <http://www.liacs.nl/home/kosters/pm/cijf/res.html>.

1. In een array `int A[n]` staan n (een `const > 0`) gehele getallen.
 - a. Schrijf een C++-functie `int tel (A,X,n)` die telt hoe vaak het gehele getal `X` in `A` voorkomt.
 - b. We nemen vanaf nu aan dat er precies drie verschillende getallen in `A` voorkomen. Schrijf een C++-functie `void welke3 (A,een,twee,drie,n)` die deze getallen in `een`, `twee` en `drie` teruggeeft, waarbij `een < twee < drie`.
 - c. Schrijf een functie `void sorteer (A,n)` die `A` oplopend sorteert. Roep hiertoe de functies van **b** en daarna ($2\times$) **a** aan, en vul `A` dan met de juiste waardes.
 - d. Schrijf de functie `void busorteer (A,n)` die `A` oplopend sorteert met *bubblesort*.
 - e. Leg uit of de methodes van **c** en **d** lineair ($O(n)$) of kwadratisch ($O(n^2)$) zijn.

2.a. 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. Gegeven een C++-programma met daarin de volgende functies:

```
int victor (int x); // prototype
void rolf (int a, int b) {
    int tmp; z--;
    if ( a < 3 ) cout << "Q " << victor (a) << endl;
    else { tmp = a; a = b; b = tmp; } //else
    cout << "R " << a << ", " << b << ", " << y << " en " << z << endl;
} //rolf
int victor (int x) {
    int hulp = 2011;
    while ( z >= 0 ) { hulp = x - 2; rolf (hulp,x); } //while
    cout << "V " << x << ", " << y << " en " << z << endl;
    return x + 32;
} //victor
```

Verder zijn de globale variabelen `y` en `z` gegeven (beide van type `int`). Voordat de functie `victor` wordt aangeroepen hebben zij de waarde 10 en 3, respectievelijk. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
y = victor (y); cout << y << " en " << z << endl;
```

- c.** Als **b**, maar nu met een `&` (“ampersand”) bij alle parameters van de functies (ook bij het prototype).
- d.** Als in het programma ergens `y = victor (y+z)`; staat, compileert het programma dan nog? Onderscheid de gevallen met en zonder `&`, zie **b** en **c**.
- e.** Waarom is het prototype nodig?

3. Gegeven is een m bij n (beide `const > 0`) array `schat`, gevuld met getallen ≥ 0 ; `schat[i][j]` geeft het aantal schatten op punt (i, j) van ons schateiland aan.

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

a. Schrijf een C++-functie `int maxi (schat,p,q)` die het punt (p, q) oplevert waarvoor geldt dat het samen met zijn direct aangrenzende horizontale en verticale burens de meeste schatten bevat. Dit maximale aantal moet teruggegeven worden. Neem aan dat zo'n punt uniek is. In het voorbeeld: $p = 0$ en $q = 3$, met $1 + 5 + 9 + 0 = 15$ schatten.

b. Schrijf een C++-functie `bool vol (schat,r)` die `true` oplevert precies dan als ieder punt van de r -de kolom van `schat` minstens één schat bevat. In het voorbeeld geldt dit alleen voor $r = 2$.

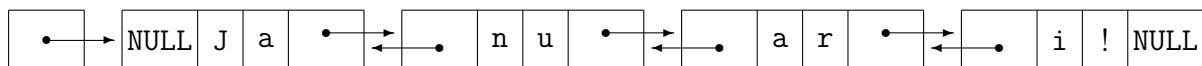
c. Schrijf een C++-functie `int langste (schat)` die berekent wat de lengte is van een langste horizontaal aaneengesloten serie punten, waarbij ieder punt een oneven aantal schatten heeft. In het voorbeeld 3 (uit de bovenste rij).

4. Gegeven is het volgende type:

```
class duo { public: char c1; char c2; duo* vorig; duo* volg; };
```

Met behulp hiervan worden lijstjes met tweetallen karakters opgebouwd. Het veld `volg` bevat een pointer naar het volgende duo-object, en de `vorige`-pointer wijst naar het vorige duo-object. Een voorbeeld (lijst van type `duo*`):

lijst



a. Schrijf een C++-functie `verwissela (lijst)` die de eerste twee duo-objecten uit de lijst (met `lijst` van type `duo*` als ingang) netjes verwisselt, mits er minstens twee zijn. Let op: verwissel pointers, en niet inhoud!

b. Schrijf een C++-functie `verwisselb (lijst)` die de *inhouden* (de karakters dus) van de eerste twee duo-objecten uit de lijst netjes verwisselt, mits er minstens twee zijn én er in het eerste object minstens één hoofdletter staat (zoals in het voorbeeld).

c. Schrijf een C++-functie `voegtoe (lijst,cc1,cc2)` die een nieuw duo-object met karakters `cc1` en `cc2` erin vooraan de lijst toevoegt.

d. 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? Leg duidelijk uit.

e. Schrijf een C++-functie `verwijder (lijst,k)` die de eerste k duo-objecten uit de lijst netjes verwijdert. Hierbij is de integer $k \geq 0$. Als er minder dan of evenveel als k duo-objecten in de lijst zitten, worden deze allemaal verwijderd.