

# Tentamen Programmeermethoden

## Maandag 7 januari 2008, 14.00–17.00 uur

### Universiteit Leiden — Informatica

Bij alle te schrijven functies moeten de variabelen (constanten uitgezonderd) in de heading of als locale variabele voorkomen; vul zelf de headings goed in. De opgaven tellen alle vier even zwaar mee. Succes! Cijfers: <http://www.liacs.nl/home/kosters/pm/res07.txt>.

1. In een array `int A[n]` stoppen we  $n$  (een `const > 0`) positieve gehele getallen.
  - a. Schrijf een C++-functie `int cijfers (A,i)` die het aantal cijfers van het  $i$ -de element van `A` teruggeeft. Stel dat `A[i]` 12354 is, dan hebben we 5 cijfers.
  - b. Schrijf een C++-functie `int bijelkaar (A,x,vanaf)` die het volgende doet. We bekijken `A` vanaf het `vanaf`-de element. Alle optredens van getallen met  $x$  cijfers in dat gedeelte van `A` worden naast elkaar gezet. Dit gebeurt door het (eventuele) eerste optreden van zo'n getal vanaf het `vanaf`-de element met het `vanaf`-de element te verwisselen, het (eventuele) volgende optreden met de rechter buurman daarvan, etcetera. De functie moet het aantal gevonden getallen met  $x$  cijfers retourneren. Bijvoorbeeld: 4 24 7 6 23 555 555 25, met  $x$  gelijk aan 2 (twee cijfers) en `vanaf` gelijk aan 3, wordt 4 24 7 23 25 555 555 6; de eerste drie elementen van `A` worden hier dus niet bekeken; return-waarde: 2. Gebruik a.
  - c. Schrijf een C++-functie `void allemaal (A)` die de getallen met één cijfer uit `A` vooraan zet, gevolgd door de getallen met twee cijfers, enzovoorts. Gebruik b.
  - d. Schrijf een C++-functie `void sorteer (A)` die `A` olopend sorteert met behulp van *bubblesort*.
  - e. De methode van d lost het probleem van c ook op. Welke methode is efficiënter? Een antwoord met "... $n^2$ ..." erin zou mooi 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 twee functies:

```
int afth (bool ako, int x); // prototype
int arnon (bool ako, int x) {
    if ( ! ako ) x = afth (ako,x); else {
        for ( y = 1; y < x; y++ ) { x = x - y; z += 1000; } }
    cout << x << ", " << y << " en " << z << endl; return x;
} //arnon
int afth (bool ako, int x) {
    int y = x + 1; if ( ako ) x = arnon (ako,x); else x = x + y;
    cout << x << ", " << y << " en " << z << endl; return x;
} //afth
```

Verder zijn de globale variabelen `x`, `y` en `z` gegeven (alle van type `int`). Voordat de functie `afth` wordt aangeroepen hebben zij de waarde 1, 5 en 8 respectievelijk. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
z = afth (true,y); cout << x << ", " << y << " en " << z << endl;
```

- c. Als **b**, maar nu staat er een `&` bij de parameters `x` (drie maal).
- d. Wat gebeurt er bij **b** (dus *zonder* de `&`'s), als we het uitroepteken (!) in `arnon` weglaten?
- e. Mag ergens de aanroep `arnon (afth (x-y,y-z),afth (z-x,x))` staan?

**3.** Gegeven is een  $m$  bij  $n$  (beide `const > 0`) array `nurikabe`, gevuld met gehele getallen. Elk getal  $> 0$  stelt een deel van een eiland voor, en een `0` staat voor water. Hiermee worden *Nurikabe* puzzels gemaakt.

6 6 0 0	OK	6 7 6	FOUT
6 0 0 8		6 0 0	
0 8 8 8		0 8 8	

**a.** Schrijf een functie `bool controle (nurikabe)` die controleert dat als er ergens eilandgedeeltes horizontaal of verticaal aan elkaar grenzen, deze hetzelfde nummer hebben.

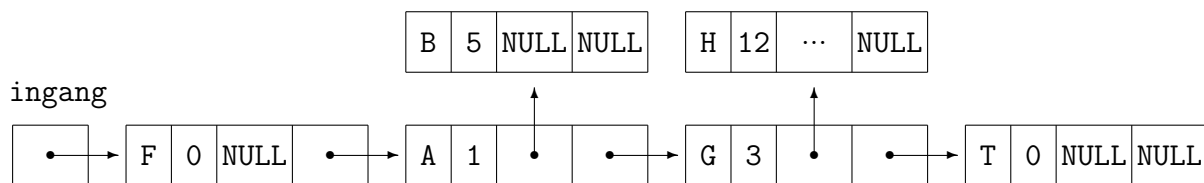
**b.** Schrijf een C++-functie `void meer (nurikabe, i, j)` die — mits op een gegeven positie  $(i, j)$  water zit — alle (eventueel in meerdere stappen) horizontaal/verticaal van daaruit bereikbare water-locaties, inclusief zichzelf, vult met `-1`. Die locaties noemen we samen een *meer*. Tip: behandel herhaald (hoe vaak maximaal?) burens van vakjes die al met `-1` gevuld zijn. Het hoeft niet efficiënt. Bij het linker voorbeeld, met  $i = 0$  en  $j = 3$ , worden vier nullen door een `-1` vervangen, alleen de nul links onder blijft.

**c.** Schrijf een C++-functie `int hoeveelmeer (nurikabe)` die bepaalt hoeveel meren er zijn. Gebruik **b**. In het linker voorbeeld zijn er 2 meren. Herstel na afloop weer de originele waarden in het array `nurikabe`.

**4.** Gegeven is het volgende type:

```
class info { public: char letter; int aantal; info* boven; info* rechts; };
```

Met behulp hiervan worden rijtjes (lijstjes) met letter-getal combinaties opgebouwd. Het veld `rechts` bevat een pointer naar het ernaast gelegen `info`-object (of `NULL`), het veld `boven` een pointer naar een lijst met `aantal` (wellicht `NULL/0`) objecten, verbonden via de `boven`-pointers (van die objecten is de `rechts`-pointer `NULL`). Een voorbeeld (`ingang` van type `info*`), waarbij `boven` het object met `H` erin nog twee objecten liggen (het object met `G` erin heeft immers 3 als `aantal`):



**a.** Schrijf een C++-functie `verwissel (ingang)` die de `letter`-inhouden van de twee eerste objecten — indien aanwezig — in de alfabetisch juiste volgorde zet. In het voorbeeld moeten de `F` en de `A` verwisseld worden.

**b.** Schrijf een C++-functie `voegtoe (ingang, let, get)` die een nieuw object met `aantal` `get`  $\leq 1$  en `letter` `let` (een hoofdletter ongelijk aan `Z`) erin vooraan de structuur (met `ingang` van type `info*` als `ingang`) toevoegt. Als `get` 1 is, moet ook het erboven liggende object worden aangemaakt, met de letter die alfabetisch op `let` volgt erin en `aantal` gelijk aan 0.

**c.** Schrijf een C++-functie `verwijder (ingang)` die het eerste object uit de lijst (met `ingang` van type `info*` als `ingang`) verwijdert. Denk aan de lege lijst. Neem aan dat de `aantal`-waarde van het te verwijderen object maximaal 1 is, en verwijder zo het er is ook het bovenliggende object.

**d.** In de functies bij **a**, **b** en **c** staat in de heading de parameter `ingang`. 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 `int desom (ingang)` die de som uitrekent van alle voorkomende `aantal`-velden.