

# Tentamen Programmeermethoden

## Vrijdag 1 april 2005, 14.00–17.00 uur

### Universiteit Leiden — Informatica

Bij alle te schrijven functies moeten de variabelen in de heading of als lokale variabele voorkomen (niet stiekem globale variabelen gebruiken). De opgaven tellen alle vier even zwaar mee. Veel succes! Cijfers: <http://www.liacs.nl/home/kosters/pm/res04.txt>.

1. Gegeven `const int n = 1000;`, en een array `A` met `n` verschillende gehele getallen (`int A[n]`). Geef de functies zelf de juiste headings (met type e.d.).

a. Schrijf een C++-functie `int aantal (A, n)` die bepaalt hoeveel buurparen `A[i]` en `A[i+1]` ( $0 \leq i < n - 1$ ) er zijn waarbij de een positief ( $\geq 0$ ) is en de ander negatief ( $< 0$ ).

b. Schrijf een C++-functie `void wissel (A, i, j)` die `A[i]` en `A[j]` verwisselt. Neem aan dat  $0 \leq i, j < n$ .

c. Schrijf een C++-functie `void bubblesort (A, n)` die het array `A` *aflopend* sorteert met behulp van *bubblesort*.

d. Schrijf nu een C++-functie `void negpos (A, n)` die alle negatieve elementen vooraan en alle positieve elementen achteraan zet. Programmeer één doorgang door het array (botweg sorteren is dus niet goed) met behulp van een index `i` die van links naar rechts loopt en een index `j` die van rechts naar links loopt. Roep indien nodig steeds de functie van **b** aan.

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 heb je ook nog *formele* en *actuele* parameters. Leg deze zes begrippen duidelijk uit.

b. Gegeven een C++-programma dat de volgende functies bevat:

```
void een (int a, int b) {
    int i; int x = 5; for ( i = 1; i <= a; i++ ) { b++; x--; } //for
    cout << a << ", " << b << ", " << i << ", " << x << endl;
} //een
void ander (int b, int a) {
    int i; for ( i = 1; i <= x; i++ ) { een (a, b); x--; } //for
    a--; cout << a << ", " << b << ", " << i << ", " << x << endl;
} //ander
```

Laten verder de globale variabelen `x`, `y` en `z` (alle van type `int`) gegeven zijn. Voor aanroep van de functie `ander` hebben zij de waarden 3, 4 en 3 respectievelijk. Wat is dan de uitvoer van (leg je antwoord duidelijk uit):

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

c. Dezelfde vraag als **b**, maar nu staat bij *elke* parameter een `&` (vier maal dus).

d. Wat gaat er fout bij de aanroep `ander (x, x)`, weer in het geval van **c**?

e. Wat is in het algemeen de waarde die in `y` en `z` zit na aanroep `een (y, z)`, uitgedrukt in de oorspronkelijke `y`, `z` en `x`? (Wederom met de `&`'s erbij.) Evenzo voor `ander (y, z)`.

3. Gegeven is een `n` bij `n` array `eennul`, gevuld met nullen en enen (dus `int eennul [n] [n]`, waarbij array-elementen 0 of 1 zijn). Voorbeelden met `n = 3`:

0 1 0	1 0 1
1 0 1	1 0 0
0 1 0	0 1 0

- a. Schrijf een C++-functie `int tel (eennul)` die het totaal aantal nullen in `eennul` telt.
- b. Schrijf een C++-functie `void som (eennul, i, j, rijSom, kolomSom)` die de som der elementen in rij `i` berekent en deze via `rijSom` doorgeeft. Analoog voor `kolomSom`.
- c. Schrijf een C++-functie `bool kruis (eennul)` die nagaat of op de twee hoofddiagonalen nullen staan en in de rest van het array enen. Voor het linkervoorbeeld zou de functie `true` opleveren, voor het rechtervoorbeeld `false`.
- d. Schrijf een C++-functie `int nultrap (eennul)` die nagaat hoeveel stappen je kan doen voordat je een 1 tegenkomt als je volgens onderstaand recept door het array loopt. We beginnen linksboven (in `eennul[0][0]` dus; wat daar staat doet er niet toe). Doe nu steeds het volgende: ga afwisselend één positie naar rechts als daar (rechts dus) een 0 staat, en vervolgens één positie naar beneden als daar een 0 staat. Stop zodra je een 1 tegenkomt of je in `eennul[n-1][n-1]` terechtkomt. Gebruik een while-loop. Voor het linkervoorbeeld kun je 0 stappen doen, voor het rechtervoorbeeld 4.

4. Gegeven is het volgende type:

```
class resultaat { public:
    int vakcode; int jaar; int cijfer;
    resultaat* volgende; resultaat* vorige;
}; // resultaat
```

Hiermee kan een dubbelverbonden lijst worden gemaakt die alle behaalde cijfers van een student bevat. Elk vakje bevat een code voor het vak, het jaar waarin het tentamen gedaan is en het behaalde cijfer. We nemen aan dat de lijst chronologisch gesorteerd is: de meest recent behaalde cijfers vooraan, de oudste achteraan. Een vak wordt maximaal één keer per jaar gedaan. Het veld `volgende` bevat steeds een pointer naar het er direct “rechts” naast gelegen resultaat, het veld `vorige` een pointer naar het er “links” van gelegen resultaat — als die bestaan, anders NULL. Een voorbeeld:

`begin`  $\longrightarrow$  NULL 253 2005 6  $\longleftarrow$  581 2004 8  $\longleftarrow$  253 2004 5 NULL  $\longleftarrow$  `eind`

Hierbij zijn `begin` en `eind` pointers van type `resultaat*`. De pointer `begin` wijst naar het begin van de lijst, de pointer `eind` naar het eind daarvan.

- a. Het cijfer van een student voor een zeer recent gedaan tentamen moet veranderd worden. Bekend is dat het het tweede of het derde cijfer uit de lijst is. Schrijf nu een C++-functie `verander (begin, code, punt)` die het cijfer voor het betreffende tentamen (vakcode `code`) verandert in `punt`.
- b. Schrijf een C++-functie `voegtoe (begin, code, year, punt)` die het zojuist behaalde resultaat (`punt`) voor het tentamen van het vak met vakcode `code` en jaar `year` vooraan in de lijst zet. Denk aan het eventuele bijwerken van een `vorige`-pointer. Neem aan dat de lijst niet leeg is.
- c. Schrijf een C++-functie `doeweg (begin, eind)` die het laatste resultaat uit de lijst verwijdert. Neem aan dat de lijst niet leeg is.
- d. Schrijf een C++-functie `gooiweg (begin, eind, year)` die alle resultaten die dateren van voor het jaar `year` uit de lijst weghaalt. Gebruik hierbij herhaald **c**.
- e. In de functies bij **a**, **b**, **c** en **d** staat in de heading de parameter `begin` en/of `eind`. 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? Motiveer je antwoord.