

Tentamen Programmeermethoden

Woensdag 23 februari 2011, 14.00–17.00 uur

Universiteit Leiden — Informatica

Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de heading of als lokale 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 telkl (A,X,n)` die telt hoeveel getallen in `A` kleiner dan of gelijk aan het gehele getal `X` zijn.

NB Geef hier (en ook in de andere opgaven) de compleet ingevulde heading van de functie!

b. Schrijf een C++-functie `int tel (A,X,n)` die telt hoe vaak het gehele getal `X` in `A` voorkomt. Hierbij mag `A` alleen benaderd worden met de functie van **a**. Tip: doe dit $2\times$.

c. Schrijf een C++-functie `bool perm (A,B,n)` die precies dan `true` oplevert als de arrays `A` en `B` precies dezelfde getallen bevatten, en ieder voorkomend getal ook even vaak. Tip: gebruik de functie van **b**.

d. Schrijf een functie `void sorteer (A,B,n)` die `A` oplopend naar `B` sorteert (en `A` onveranderd laat). Roep hiertoe herhaald de functies van **a** en **b** aan, en vul `B` dan met de juiste waardes.

e. Is de methode van **d** slechter dan, vergelijkbaar met, of beter dan bubblesort? Leg uit.

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 Sarah (int x, int y) {
    int a, iets = 0;
    for ( a = y; a > 0; a-- ) {
        iets += x; y--; cout << "S " << a << " " << iets << " " << y << endl; }
    return iets;
} //Sarah
int Jan (int a, int b) {
    int x = 1; b += x; a--; a = Sarah (b,a) + Sarah (a,a);
    cout << x << " " << a << " " << b << endl; return a * (2-x);
} //Jan
```

Verder zijn de globale variabelen `x` en `y` gegeven (beide van type `int`). Voordat de functie `Jan` wordt aangeroepen hebben zij de waarde 4 en 2, respectievelijk. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
cout << Jan (x,y); cout << " " << x << " en " << y << endl;
```

c. Geef een functie `int John (int a, int b)` die dezelfde return-waarde oplevert als `Jan` voor alle mogelijke waardes van de parameters, maar die uit slechts één `return`-statement bestaat, en waarin `Sarah` niet meer wordt aangeroepen.

d. Als **b**, maar nu met een `&` (“ampersand”) bij alle parameters van de functies. Geef alle mogelijke uitvoeren, en leg uit waarom er hierbij verschillende waardes mogelijk zijn.

e. Als in het programma ergens `x = Jan (y-x,y+x)`; staat, compileert het programma dan nog? Onderscheid de gevallen met en zonder `&`, zie **b** en **d**.

3. Gegeven is een m bij n (beide `const > 0`) array `puzzel`, gevuld met *verschillende kleine letters*. In het voorbeeld rechts geldt $m = 3$ en $n = 6$.

```
q w e r t y
a s d f g h
z x c v b n
```

a. Schrijf een C++-functie `opvolger (puzzel, letter, p, q)` die het coördinaten-paar (p, q) oplevert waarin de alfabetisch eerste letter uit `puzzel` zit die op de char `letter` volgt. Mocht `letter` de alfabetisch laatste zijn, dan moeten p en q beide -1 worden. In het voorbeeld: `letter = 't'` geeft $p = 2$ en $q = 3$ (de 'v').

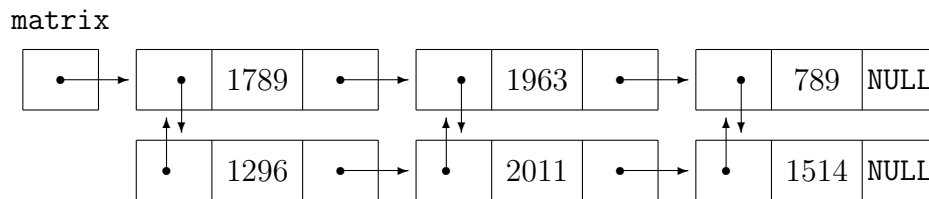
b. Schrijf een C++-functie `alpha (puzzel)` die de letters uit de `puzzel` in alfabetische volgorde afdrukt. Gebruik de functie van **a**. In het voorbeeld: `abcdefghijklmnopqrstvwxyz`.

c. Schrijf een Booleaanse C++-functie `komtvoor (puzzel, een, twee)` die bepaalt of de char's `een` en `twee` als horizontale of verticale burens voorkomen in `puzzel`. In het voorbeeld: `true` voor 'd' en 'e', maar `false` voor 'q' en 'y'.

4. Gegeven is het volgende type:

```
class element { public: int jaar; element* onderboven; element* rechts; };
```

Met behulp hiervan worden rechthoekige matrices met twee rijen (verschillende) jaartallen opgebouwd. Het veld `rechts` bevat een pointer naar het er rechts naast gelegen `element`-object, en de `onderboven`-pointer wijst naar het er onder of er boven gelegen `element`-object, dat altijd bestaat. Een voorbeeld (`matrix` van type `element*`) voor een 2×3 -matrix:



a. Schrijf een C++-functie `voegtoe (matrix, jaar1, jaar2)` die *twee* nieuwe `element`-objecten met jaartallen `jaar1` en `jaar2` erin als linkerkolom vooraan de `matrix` (met `matrix` van type `element*` als ingang) toevoegt. In het voorbeeld zijn zojuist 1789 en 1296 voorgevoegd.

b. Schrijf een C++-functie `verwissel (matrix)` die het *jaartal* van het `element`-object linksboven uit de `matrix` verwisselt met het grootste van zijn rechter- en onderbuur, mits deze bestaan. Als alleen de onderbuur bestaat (en dat is zo als de `matrix` niet leeg is), moet daarmee gewisseld worden. In het voorbeeld moet 1789 met 1963 verwisseld worden.

c. Schrijf een C++-functie `verwijder (matrix)` die de twee `element`-objecten uit de eerste kolom van de `matrix` verwijdert. Houd rekening met een lege `matrix`.

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 `int verschil (matrix)` die berekent wat in de `matrix` het grootste absolute verschil is tussen een jaartal uit de bovenste rij en zijn directe onderbuur. Neem aan dat de `matrix` niet leeg is. In het voorbeeld is het antwoord $1514 - 789 = 725$.