Tentamen Programmeermethoden
Vrijdag 27 maart 2009, 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/res08.txt`.

1. In een array `int A[n]` stoppen we `n` (een `const` > 0) gehele positieve (> 0) getallen.
a. Een *run* is een rijtje dezelfde getallen die direct naast elkaar staan. Schrijf een C++-functie `int larun (A,n)` die de lengte teruggeeft van de langste run die in `A` voorkomt. Voor `5 5 1 5 5 7 7 7 2` is het antwoord `3`, voor `8 6 8` is het `1` (!), voor `9 9 2` is het `2`.
b. Schrijf een boolese C++-functie `meer (A,n)` die precies dan `true` teruggeeft als in `A` minstens 2 runs met hetzelfde getal voorkomen, zoals bij de twee eerste array's van **a**. De derde geeft `false`.
c. Schrijf een C++-functie `schuif (A,n)` die elke run door één exemplaar van het betreffende getal vervangt, en dit naar het begin aanschuift. De resterende array-elementen moeten `0` worden. In het eerste voorbeeld van **a** wordt het array `5 1 5 7 2 0 0 0 0`.
NB Dubbele loops zijn niet nodig.
d. Schrijf een C++-functie `sort (A,n)` die `A` oplopend sorteert met behulp van *bubblesort*.

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 doedit (int v, int w) {
  if ( v > w ) { v = v - w; k++; }//if
  else { w = w - v; k++; }//else
  cout << v << "," << w << "," << k << endl; return v + w;
}//doedit
int wiebenik (int a, int b) {
  int s = 0, temp;
  while ( a != 0 && b != 0 && k < 4 ) {
    temp = a + b; s += doedit (a,b);
    if ( temp != a + b ) k = 0; }//while
  cout << a << "," << b << "," << s << endl; return s;
}//wiebenik
```

Verder zijn de globale variabelen `p`, `q` en `k` gegeven (alle van type `int`). Voordat de functie `wiebenik` wordt aangeroepen hebben zij de waarde `17`, `7` en `0`, respectievelijk. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
cout << wiebenik (p,q) << endl; cout << p << "," << q << "," << k << endl;
```

c. Als **b**, maar nu met een `&` bij de vier parameters van de functies.
d. Neem aan dat `a` en `b` beide positief zijn. Leg uit waarom de `while`-loop in de functie `wiebenik` altijd stopt, zowel met als zonder de `&`'s in `doedit`.
e. Stel dat na aanroep van `wiebenik` `p` of `q` de waarde 0 heeft (weer met de vier `&`'s erbij, als in **c**); wat heeft de waarde van de ander dan (in het algemeen) te maken met die van de oorspronkelijke `p` en `q`?

3. Gegeven is een `m` bij `n` (beide `const` > 0) array `schaak`, gevuld met schaakstukken `1` (koning) en `2` (dame), terwijl `0` staat voor een leeg veld; `schaak[i][j]` stelt het stuk op positie $(\texttt{i}, \texttt{j})$ voor.

```
0 1 0 2 1 0 0
0 0 0 0 1 0 0
2 0 0 0 0 0 0
```

Een koning kan in één zet alle 8 horizontaal, verticaal en diagonaal direct aangrenzende velden bereiken (een paar minder aan de rand). Een dame kan in één zet alle velden in haar rij, kolom en twee diagonalen bereiken. We spreken af dat een dame zelfs over andere stukken heen kan springen.

a. Schrijf een C++-functie `aanvalkoning (i,j,p,q)` die precies dan `true` oplevert als een koning op plek $(\texttt{i}, \texttt{j})$ plek $(\texttt{p}, \texttt{q})$ in één keer kan bereiken. Neem aan dat $(\texttt{i}, \texttt{j}) \neq (\texttt{p}, \texttt{q})$.

b. Idem, maar nu voor de dame: een functie `aanvaldame`.

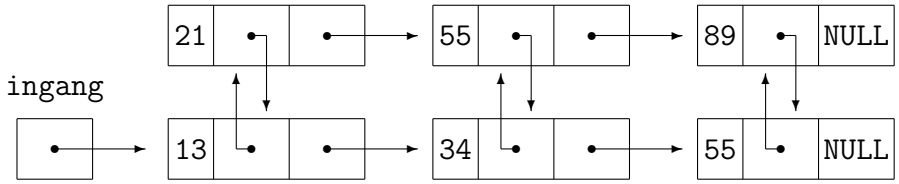
c. Schrijf een boolese C++-functie `dame (schaak,i,j)` die `true` oplevert precies dan als er op $(\texttt{i}, \texttt{j})$ een dame staat die geen ander stuk direct kan slaan. In het voorbeeld geeft `dame (schaak,2,0)` als antwoord `true`, en `dame (schaak,0,3)` levert `false`. Gebruik **b**.

d. Schrijf een C++-functie `int aantal (schaak)` die bepaalt hoeveel velden er door minstens één stuk in één zet bereikt kunnen worden. Velden waar een stuk staat tellen ook mee. In het voorbeeld: 20 — alleen veld $(1, 6)$ niet. Gebruik **a** en **b**.

4. Gegeven is het volgende type:

```
class getal { public: int info; getal* andere; getal* volg; };
```

Met behulp hiervan worden parallelle lijstjes met getallen opgebouwd. Het veld `volg` bevat een pointer naar het horizontaal volgende `getal`-object. De `andere`-pointer wijst verticaal naar de parallelle lijst. Steeds geldt dat als het onderste object bestaat, het bovenste er ook is. Een voorbeeld (`ingang` van type `getal*`):



a. Schrijf een C++-functie `verwissel (ingang)` die de `info`-inhouden van de twee meest linker objecten — indien aanwezig — verwisselt. In het voorbeeld moeten `13` en `21` verwisseld worden.

b. Schrijf een C++-functie `voegtoe (ingang,cf1,cf2)` die een nieuw tweetal `getal`-objecten met `info` `cf1` en `cf2` erin vooraan de structuur (met `ingang` van type `getal*` als ingang) toevoegt. Bij de voorbeeldlijst is zojuist `voegtoe (ingang,13,21)` aangeroepen.

c. Schrijf een C++-functie `verwijder (ingang)` die de twee boven elkaar gelegen `getal`-objecten opruimt waarmee de lijst (met `ingang` van type `getal*` als ingang) begint, mits de `info`-velden verschillende getallen bevatten. Denk aan de lege lijst. In het voorbeeld moeten de objecten met `13` en `21` verwijderd worden.

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. Neem aan dat door een ongelukje steeds van elk onder elkaar liggend tweetal objecten een van de twee horizontale `volg`-pointers `NULL` is geworden. Schrijf een C++-functie `repareer (ingang)` die al deze pointers herstelt.