

Tentamen Programmeermethoden

Donderdag 3 januari 2002, 10.00–13.00 uur

Universiteit Leiden — Informatica

Bij alle te schrijven functies moeten de variabelen in de heading voorkomen (niet stiekem globale variabelen gebruiken). De opgaven tellen alle vier even zwaar mee. Veel succes!

1. We hebben een array `A` (`double A[n]`, met `const int n = 1234;`) met `n` getallen.
 - a. Geef een C++-functie `void wissel (A,i,j)` die de waarden van de array-elementen `A[i]` en `A[j]` verwisselt (bij vaste `i` en `j` met $0 \leq i, j \leq n-1$).
 - b. Neem aan dat de array-elementen `A[i+1]`, `A[i+2]`, ..., `A[j]` oplopend gesorteerd zijn. Geef een C++-functie `void bergop (A,i,j)` die `A[i]` op de juiste plek opbergt tussen `A[i+1]`, `A[i+2]`, ..., `A[j]`, waarbij eventuele getallen kleiner dan `A[i]` één plaatsje naar voren gewisseld worden. Neem aan dat $0 \leq i < j \leq n-1$.
 - c. Geef een C++-functie `void invoegsorteer (A,n)` die het array `A` oplopend sorteert door herhaald de functie `bergop` aan te roepen.
 - d. Hoeveel vergelijkingen tussen `double`'s doet het algoritme van `c` *minstens*, voor algemene `n`? Geef een voorbeeld wanneer dit optreedt.
 - e. Als `d`, maar nu *hoogstens*. Geef ook weer een voorbeeld.
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 aan de hand van een *klein* voorbeeld duidelijk uit.
- b. Een zeker C++-programma bevat de volgende programmaregels:

```
int draai (int & i, bool & b) {
    i = -i; b = !b; if ( b ) i++;
    cout << i << b; return i+1;
} // draai
void jaar2002 (int & i, int grens) {
    int k; bool ja = true;
    for ( k = 1; k <= grens; k++ ) cout << draai (i,ja) << endl;
    cout << i << k << grens << endl;
} // jaar2002
```

Wat levert op (met globale variabelen `a` en `grens` van type `int`):

```
a = 5; grens = 4;
jaar2002 (a,grens); cout << a << grens << endl;
```

Wat wordt er afgedrukt? Geef hierbij uiteraard uitleg.

- c. Als `b`, maar nu zonder de drie `&`'s in de headings van `draai` en `jaar2002`.
- d. De situatie is weer als bij `b`, dus met de drie `&`'s erbij. Stel dat `jaar2002 (a,grens)` aangeroepen wordt in een situatie waarin de parameter `a` de waarde 0 heeft. Welke waarde moet de parameter `grens` hebben opdat na afloop van de aanroep `a` de waarde 2002 heeft?
- e. Geef een korte C++-functie `void nieuw (int & i, int grens)` die na afloop in de variabele `i` dezelfde waarde heeft als het geval zou zijn bij de oorspronkelijke functie `void jaar2002 (int & i, int grens)`, voor iedere waarde van `i` en `grens`.
- f. Stel dat je in de functie `draai` de functie `jaar2002` zou willen aanroepen. Mag dat, en zo nee, wat moet je doen om het wel te mogen laten kunnen?

3. Voor het spelletje onderzeeërtje gebruiken we een m bij n array A met booleans (voorbeeld met $m = 2$ en $n = 5$; `true` betekent: hier ligt een deel van een schip):

```
true  false false true  true
false false true  false true
```

a. Schrijf een C++-functie `int hoelangrij (A,i,m)` die de lengte van de langste serie aaneengesloten `true`'s uit de i -de rij van het m bij n array A oplevert — de lengte van het langste schip in deze rij dus. Met $i=0$ zou het antwoord 2 zijn, en met $i=1$ zou het antwoord 1 zijn.

b. Schrijf een C++-functie `bool controle (A,m)` die checkt of het m bij n array A een correcte opstelling van schepen is; hiertoe moet worden gecontroleerd of er nooit twee delen van een schip zijn die elkaar *diagonaal* raken. In het voorbeeld levert de functie `false` op: diverse boten raken elkaar diagonaal.

c. Schrijf een C++-functie `bool duikboten (A,m)` die bepaalt of het array alleen bestaat uit *duikboten*; een duikboot bestaat uit één `true`, waaraan horizontaal, verticaal en diagonaal geen andere `true`'s grenzen (in het voorbeeld ligt er een linksboven). Gebruik `a` en `b`, waarbij — zonder deze te hoeven schrijven — ook de kolomversie `hoelangkolom (A,j,m)` van de functie uit `a` gebruikt mag worden.

4. Gegeven is het volgende type:

```
class pakket { public: // een struct mag ook
    int info; pakket* volgende; pakket* vorige;
}; // pakket
```

Met behulp hiervan kunnen rijtjes getallen (pakketten) worden opgebouwd. Het veld `volgende` bevat steeds een pointer naar het er direct “rechts” naast gelegen pakket, het veld `vorige` bevat een pointer naar het er “links” van gelegen pakket — als die bestaan, anders NULL. Een voorbeeld: `ingang` \rightarrow NULL 1234 \leftrightarrow 5678 \leftrightarrow 7777 NULL

Hierbij is `ingang` van type `pakket*`. Voor het derde pakket, met 7777 erin, wijst de pointer `vorige` naar het pakket met 5678 erin; de `volgende`-pointer is daar NULL.

a. Schrijf een C++-functie `voegtoe (ingang,getal)` die een nieuw pakket met `getal` erin vooraan de structuur (met `ingang` van type `pakket*` als `ingang`) toevoegt. Denk aan het eventuele bijwerken van een `vorige`-pointer.

b. Schrijf een C++-functie `verwijder (ingang)` die het eerste pakket uit de lijst (met `ingang` van type `pakket*` als `ingang`) verwijdert indien in dat vakje een getal groter dan 1000 zit. Denk aan de lege lijst. Denk aan het eventuele bijwerken van een `vorige`-pointer.

c. Schrijf een C++-functie `verwissel (ingang)` die de `info`-velden in de twee eerste pakketten onderling verwisselt — indien deze vakjes bestaan (en anders niets doet).

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 ook uit wat er bij deze twee situaties precies gebeurt tijdens executie van de betreffende functie.

e. Een grappenmaker besluit alle `vorige`-pointers naar de *voorvorige* vakjes te laten verwijzen — NULL als zo'n vakje niet bestaat. In het voorbeeld zou de `vorige`-pointer vanuit 7777 moeten gaan wijzen naar het eerste vakje (met 1234 erin). Schrijf een C++-functie `grap (ingang)` die dit doet voor de structuur met `ingang` als `ingang`.

Zodra het tentamen is nagekeken wordt dit vermeld op het bord in de hal, zo rond 21 januari.