

Tentamen Programmeermethoden

Vrijdag 6 augustus 1999, 9.00–12.00 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. Gegeven zijn `const int n = 100;` en een array `A` met een rij van `n` onderling verschillende getallen: `int A[n];`.

a. Schrijf een `void` C++-functie `Wissel (A,i,j)` die het `i`-de en `j`-de array-element van `A` verwisselt; neem aan dat $0 \leq i,j < n$.

b. Schrijf een `void` C++-functie `Bergop (A,i)` die het array-element `A[i]` ($0 < i < n$) op de juiste plaats (in *dalend* gesorteerde volgorde) opbergt in het array `A`, tussen de eerste `i` getallen. Neem aan dat de getallen `A[0],A[1],...,A[i-1]` reeds dalend gesorteerd zijn. Gebruik **a**.

c. Schrijf een C++-functie `InsertionSort (A)` die het array `A` met behulp van insertion sort (invoegsorteer) dalend sorteert. Gebruik **b**.

d. Hoeveel vergelijkingen doet het algoritme van **c** ongeveer bij het dalend sorteren van de rij `1,2,...,n`?

e. Stel dat het array `A` dalend gesorteerd is. De functie `Wissel (A,i,j)` wordt nu aangeroepen met onbekende waarden van `i` en `j`. Hoeveel vergelijkingen tussen array-elementen, uitgedrukt in `i` en `j`, doet het algoritme van **c** ongeveer bij het opnieuw sorteren van deze rij?

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. Gegeven is de volgende functie:

```
int gcd (int x, int y) {
    int temp = 0;
    while ( y != 0 ) { temp = x % y; x = y; y = temp;}
    cout << x << y << temp << endl; return x; } // gcd
```

Neem aan dat de waardes van de globale variabelen `u`, `v` en `w`, alle `int`, 6, 8 respectievelijk 13 zijn. Wat gebeurt er bij `cout << gcd (u,v+w) << u << v << w << endl;`? Wat wordt er afgedrukt? Probeer duidelijke uitleg te geven.

c. We voegen twee maal een `&` toe, en wel bij beide parameters in de heading van `gcd`. We noemen de nieuwe functie `gcd2`. Beantwoord opnieuw vraag **b**, maar nu voor `cout << gcd2 (u,v) << u << v << w << endl;`.

d. Geldt altijd `gcd (r,s) == gcd (s,r)`? Neem aan dat $r,s \geq 0$, niet beide 0. Leg uit.

e. Geldt altijd `gcd2 (r,s) == gcd2 (s,r)`? Leg uit.

f. Het effect van de functie `gcd2` kan eenvoudig met behulp van `gcd` worden uitgedrukt. Zo kan de `while`-loop uit `gcd2` verwijderd worden ten koste van één of meer aanroepen naar `gcd`. Doe dit.

3. In een rechthoekig array `R (const int m = 45; const int n = 87; char R[m] [n];)` staan karakters uit `{'0','1','2','3','4','5','6','7','8','9'}`. Een voorbeeld, voor het gemak even met `const int m = 3; const int n = 4;`:

```
'8' '9' '7' '0'
'5' '6' '2' '4'
'1' '9' '4' '4'
```

a. Maak een `int` C++-functie die het aantal verschillende karakters voor de `i`-de rij oplevert — in het voorbeeld 4 voor de eerste twee rijen, of 3 voor de laatste.

b. Schrijf een `int` C++-functie die de index van de (of een) rij met de meeste verschillende karakters geeft — in het voorbeeld 0 of 1. Gebruik **a**.

c. De “telsom” beginnend op positie (i,j) wordt als volgt berekend: bepaal eerst de getalswaarde van het vakje waar je bent ('4' wordt 4, etcetera) en tel deze op bij de tot dan toe berekende som; als er diagonaal een vakje aan het huidige vakje grenst dat precies één meer is, is de telsom klaar, anders ga je naar de rechter buurman (naar de eerste van de volgende rij als je aan het eind van de vorige was, en naar linksboven als je rechtsonder was). De “telsom” stopt ook als hij 1000 of meer wordt. Uiteraard begint de som op 0. Ieder vakje heeft één, twee of vier diagonale burens. Schrijf een `int` C++-functie `telsom(R,i,j)` die de “telsom” die begint in rij `i`, kolom `j`, oplevert. In het voorbeeld, met $i = 0$ en $j = 2$, is dat $7 + 0 + 5 + 6 = 18$ (het stopt wegens buurman '7').

4. Gegeven is:

```
class Mens { public: char naam; // naam van de persoon
              int hoever; // verweg wijst hoever verder
              Mens* verweg; // wijst persoon verderop aan
              Mens* volgende; // wijst volgende persoon aan
}; // Mens
```

Met behulp hiervan kan een enkelverbonden lijst van mensen worden opgebouwd. Zo'n lijst bestaat uit vakjes met een `char` en een `int`, en twee pointers naar respectievelijk een zich `hoever` vakjes (minstens 0) verder in de lijst bevindende persoon en de volgende persoon. Als `hoever = 0` wijst `verweg` het vakje zelf aan.

a. Voor zowel de eerste als de tweede persoon van een niet-lege lijst met ingang `Groep` van type `Mens*` wordt zijn/haar naam van hoofdletter naar kleine letter omgezet, en andersom. Neem aan dat de naam een kleine letter of hoofdletter is. Schrijf een C++-functie `hOOFD(Groep)` die dit doet. Vergeet niet de situatie met slechts één persoon.

b. Schrijf een C++-functie die de eerste persoon uit de lijst `Groep` verwijdert. Als de lijst leeg is of als de door `verweg` aangewezen persoon hetzelfde heet als de eerste persoon, hoeft er niets te worden gedaan.

c. Schrijf een C++-functie `Voegtoe(Groep,naampje,afstand)` die een nieuwe persoon, geheten `naampje` (van type `char`), toevoegt vooraan de eventueel lege lijst `Groep` (van type `Mens*`). Geef ook het `verweg`-veld de juiste waarde, aannemend dat `afstand` (= `hoever`) hooguit 2 is, en dat de betreffende persoon bestaat in de lijst.

d. In de functies bij **a**, **b** en **c** staat in de heading de parameter `Groep`. 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 mogelijkheden precies gebeurt tijdens executie van de betreffende functie.

e. Schrijf een `bool` C++-functie `Controle(Groep)` die controleert of voor alle personen in de lijst met ingang `Groep` het `verweg`-veld correct de persoon `hoever` verderop aanwijst.

Zodra het tentamen is nagekeken staat dit vermeld op het bord in de hal, uiterlijk maandag 16 augustus 1999.