

Tentamen Programmeermethoden

Vrijdag 5 januari 1996, 14.00–17.00 uur

Rijksuniversiteit 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 = 1000;` en `typedef int array[n];`. Een variabele `A` van type `array` bevat een rij onderling verschillende getallen. Bij deze opgave nemen we aan dat de eerste helft en de tweede helft van het array `A` oplopend gesorteerd zijn: $A[0] < A[1] < \dots < A[n/2-1]$ en $A[n/2] < A[n/2+1] < \dots < A[n-1]$.

a. Schrijf nu een C++-functie `eenna (A)` die het opeennagrootste array-element van `A` oplevert. In de functie mogen maximaal *drie* vergelijkingen waarbij een array-element betrokken is gedaan worden.

b. Schrijf een C++-functie `sorteer (A,B)` die het array `A` oplopend gesorteerd in het array `B` opbergt. Hint: gebruik twee indices die elementen uit de eerste respectievelijk tweede helft van `A` aanduiden; kopieer steeds de kleinste van de twee betreffende array-elementen naar `B`. Denk eraan dat een van de twee helften “leeg” kan (en zal) raken!

c. Hoeveel vergelijkingen tussen array-elementen doet het `sorteer`-algoritme van b *minimaal*? Geef een rijtje getallen waarbij dit gebeurt.

d. Hoeveel vergelijkingen tussen array-elementen doet het `sorteer`-algoritme van b *maximaal*? Geef een rijtje getallen waarbij dit gebeurt.

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 zijn de volgende functies:

```
int functie (int& a, int& r) {
    a++; r = r * 3;
    return (r + 1); } // functie
void werken (int& x, int y) {
    int t = 5; x = x + functie (y,r) + t;
    y = x + 3; r--; t++;
    cout << "werken" << r << x << y << t << endl; } // werken
```

Neem aan dat de waarde van de *globale variabele* `r` van type `int` bij binnenkomst van `werken` 14 is, en dat `u` en `v` `int`'s zijn met waarde respectievelijk 17 en 2. Wat gebeurt er bij de aanroep `werken (u,v);`, gevolgd door `cout << r << u << v << endl;`? Wat wordt er afgedrukt? Probeer duidelijke uitleg te geven.

c. Idem, maar nu zonder `&` in de twee headings.

d. En wat gebeurt er als de aanroep `werken (x,x);` wordt, gevolgd door het statement `cout << r << x << endl;` (`x` een `int` met waarde 5; `&` er wel bij; `r` weer 14)?

e. Als d, maar nu met `int& y` in de heading van `werken` in plaats van `int y`. Leg uit waarom er verschillende antwoorden mogelijk zijn.

3. Gegeven zijn:

```
const int n = 40; const int m = 10;
enum boolean {False = 0, True = 1};
typedef char vierkant[n][n];
typedef char woord[m];
```

We gaan vierkanten met woorden onderzoeken. Een variabele `w` van type `woord` met `m` verschillende letters *komt voor* in een variabele `v` van type `vierkant` precies dan als je beginnende bij een vakje naar keuze, alle letters uit `w` in de juiste volgorde kunt vinden door steeds naar een van de vier directe burens (twee horizontaal, twee verticaal; bij randvakjes minder) van het vakje waar je dan bent te gaan. Neem aan dat de letters in `v` zo zijn, dat geen enkel vakje directe burens heeft met dezelfde inhoud.

- Schrijf een boolese C++-functie `komtvoor(v,w,i,j)` die precies dan `True` oplevert indien `w` in `v` voorkomt, beginnend in rij `i` en kolom `j` ($0 \leq i, j \leq n-1$).
- Schrijf een C++-functie `jaofnee(v,w)` die precies dan `True` oplevert indien `w` in `v` exact één maal voorkomt. Gebruik a.

4. Gegeven zijn de volgende types:

```
class CD { char titel; // korte naam van de CD
          CD* volgende; // wijst naar de volgende CD
          CD* vorige; // wijst naar de vorige CD
}; // CD
typedef CD* CDwijzer;
```

Met behulp hiervan kan een CD-lijst worden opgebouwd. Neem aan dat elke CD een unieke titel van type `char` heeft. Een voorbeeld met drie CD's, met CDs van type `CDwijzer`:

CDs \longrightarrow NULL 'D' \longleftrightarrow 'X' \longleftrightarrow 'G' NULL

Het vakje met 'X' erin bevat hier dus een pointer (de pijl naar links) naar het vakje met 'D' erin en een pointer (de pijl naar rechts) naar het vakje met 'G' erin.

- CD 'X' uit een lijst met ingang CDs (van type `CDwijzer`) krijgt een andere titel. Schrijf een C++-functie `anders(CDs,naam)` die de titel van 'X' verandert in `naam` (van type `char`). Neem aan dat 'X' als eerste *of* als tweede in de lijst staat.
- Schrijf een C++-functie `voegtoe(CDs,naam)` die de nieuwe CD geheten `naam` (van type `char`) toevoegt vooraan de eventueel lege lijst CDs.
- Schrijf een C++-functie die de eerste CD uit de lijst CDs verwijdert. Als de lijst leeg is hoeft er niets te worden gedaan.
- In de functies bij a, b en c staat in de heading de parameter CDs. Deze heb je al dan niet als var-parameter gedeclareerd (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.
- Schrijf een boolese C++-functie `gesorteerd(CDs)` die `True` teruggeeft precies als de lijst CDs alfabetisch oplopend *of* teruglopend gesorteerd is.

Wanneer het tentamen is nagekeken wordt dit vermeld op het bord in de hal van dit gebouw, hopelijk omstreeks 30 januari 1996.