

Tentamen Programmeermethoden

Vrijdag 18 december 2015, 14:00–17:00 uur

Universiteit Leiden — Informatica



Let op: dit (her)tentamen is bedoeld voor studenten Natuurkunde en/of Sterrenkunde.

Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de heading of lokaal voorkomen; vul zelf headings goed in. De te behalen punten (totaal 100) staan tussen haakjes bij de opgaven. Succes! Cijfers: www.liacs.leidenuniv.nl/~kosterstwa/pm/cijf/res.html.

1. (35 punten)

In een array `int A[n]` staan `n` (een `const ≥ 1`) gehele getallen > 0 .

- (4) Schrijf een Booleaanse C++-functie `gesorteerd (A,n)` die precies dan `true` oplevert als het array `A` oplopend gesorteerd is.
 - (7) Schrijf een niet-recursieve functie `int fibo (i)` die het `i`-de Fibonacci-getal uitrekent, oftewel het `i`-de getal uit de serie 1 1 2 3 5 8 13 21 ...; bijvoorbeeld het 4-de Fibonacci-getal is gelijk aan 5. Neem aan dat het gehele getal `i` minstens 0 is.
 - (6) Schrijf een Booleaanse C++-functie `fibocheck (A,n)` die precies dan `true` oplevert als het array `A` de Fibonacci-getallen bevat (`A[i]` bevat het `i`-de Fibonacci-getal), waarbij er *maximaal* twee fout mogen zijn; verder moet het array tevens oplopend gesorteerd zijn. Dus het array 1 1 2 3 5 6 13 21 moet `true` opleveren.
 - (5) Schrijf een C++-functie `sorteer (A,n)` die het array `A` oplopend sorteert. Geef bij voorkeur een eenvoudige methode.
 - (3) Hoeveel vergelijkingen tussen array-elementen doet de methode van **d**, uitgedrukt in `n`? Leg het antwoord duidelijk uit.
 - (10) Schrijf een C++-functie `int langste (A,n,gem)` die de lengte uitrekent van een langste stijgende (of beter: niet-dalende) aaneengesloten deelrij in het array `A`. Hierbij moet `gem` het op de gebruikelijke manier naar een geheel getal afgeronde gemiddelde zijn van de getallen in de betreffende deelrij. Voor het array 1 7 1 1 2 7 6 3 zou het antwoord 4 zijn, namelijk de lengte van de deelrij 1 1 2 7. En `gem` moet 3 worden (via $11/4$). (Als er meer deelrijen dezelfde maximale lengte realiseren: het gemiddelde van de laatste.)
-

2. (30 punten)

- (6) 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 *formele* en *actuele* parameters. Leg deze zes begrippen duidelijk uit.
- (6) Gegeven een C++-programma met daarin de volgende twee functies:

```
int hillary (int n, int m) {
    n--; return n+m-1; }//hillary
int donald (int n, int m) {
    int a = 3; m--; n += 2; a++;
    m = n + hillary (n,m) + hillary (m,n) + a;
    cout << a << ", " << m << ", " << n << endl; return a+n-m; }//donald
```

Verder zijn de globale variabelen `a` en `m` gegeven, beide van type `int`. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

```
a = 1; m = 4; cout << donald (m,a) << endl;
cout << a << ", " << m << endl;
```

- c. (5) Idem, maar nu met twee `&`'s in de heading van `donald`.
- d. (4) We voegen nu ook nog twee maal een `&` toe bij de parameters in de heading van `hillary`. Beantwoord opnieuw vraag **c**; leg uit waarom verschillende uitkomsten mogelijk zijn, en geef deze.
- e. (4) Als in de functie `hillary` ergens `a = donald (0,a-a)`; staat, compileert het programma dan nog? Onderscheid gevallen met en zonder `&`.
- f. (5) Geef een eenvoudige uitdrukking voor de functiewaarde van `donald (r,s)`, uitgedrukt in `r` en `s`, voor de situatie zonder `&`'s (net als bij **b**).

3. (35 punten)

Gegeven is een `m` bij `n` (beide `const > 0`) array `M` met gehele getallen ≥ 0 . Hierbij geeft `M[i][j]` het aantal mensen ter plekke (i, j) aan (met $0 \leq i < m$ en $0 \leq j < n$). Hiernaast staat een voorbeeld met `m = 4` en `n = 5`. De constanten `m` en `n` hoeven bij deze opgave niet doorgegeven te worden als parameter.

3	0	4	2	0
7	3	4	10	6
6	2	0	7	1
6	6	3	0	0

- a. (7) Schrijf een C++-functie `int maxim (M,i,j)` die het grootste getal uit `M` teruggeeft, en de locatie van die plek in het paar (i, j) stopt. In het voorbeeld: waarde 10, en $i = 1$ en $j = 3$. Als het maximum meerdere keren wordt aangenomen in `M`, geef dan het duo met de grootste waarde voor $m \cdot i + j$. Vergeet niet de heading compleet in te vullen!
- b. (8) Schrijf een C++-functie `int druk (M,maxsom)` die de “drukste rij” uit `M`, dat wil zeggen de index van de rij met de hoogste som, retourneert. In de variabele `maxsom` moet de betreffende som worden teruggegeven. Neem aan dat alle rij-sommen verschillen. In het voorbeeld: rij 1 met `maxsom` gelijk aan 30.
- c. (10) We willen paren vormen bestaande uit even grote groepen mensen uit naburige locaties. Schrijf hiertoe een C++-functie `int paren (M,ondergrens)` die uitrekent hoeveel direct horizontaal of verticaal aangrenzende (eventueel overlappende) buurparen in `M` evenveel mensen bevatten, en wel samen minstens `ondergrens`. In het voorbeeld, met `ondergrens 6`: 3, namelijk 4–4, 6–6 en 6–6 (0–0 haalt de ondergrens niet).
- d. (10) Er gaan mensen verhuizen, en wel als volgt. Zoek herhaald het grootste getal in `M` met de functie van **a**. Verplaats daar één persoon naar de (twee, drie of vier horizontaal of verticaal direct aangrenzende) burens, mits die minstens 5 kleiner zijn. Het algoritme stopt als er niemand verplaatst wordt, en dat gaat zeker een keer gebeuren. In het voorbeeld zou in de eerste stap de 10 met 2 worden verlaagd naar 8, terwijl de aangrenzende 2 en 4 met 1 worden opgehoogd; de 6 en 7 veranderen niet. Er wordt dus één persoon verplaatst naar elke buur die klein genoeg is. Schrijf een C++-functie `int verhuis (M)` die dit uitvoert, en het aantal stappen totdat het algoritme stopt uitrekent.