

Tentamen Programmeermethoden

Donderdag 31 maart 2022

14:15–17:15 uur Informatica



Universiteit
Leiden
The Netherlands

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: te zijner tijd via Brightspace/uSis.

1. (25 punten) In het array `int A[n]` staan n (een `const int > 1`) verschillende gehele getallen.
 - a. (5) Schrijf een C++-functie `int maxi (A,k)` die de array-index teruggeeft van het grootste element in het deel-array `A[0]..A[k]`, waarbij $0 \leq k < n$.
 - b. (7) Schrijf een C++-functie `flip (A,k)` die het hele deel-array `A[0]..A[k]` ($0 \leq k < n$) omdraait: wissel eerste en laatste, enzovoorts.
 - c. (8) Schrijf een C++-functie `sorteer (A,n)` die het array `A` oplopend sorteert door herhaald het grootste element (gebruik de functie van **a**) uit het ongesorteerde voorste deel van het array achteraan te zetten (gebruik tweemaal de functie van **b**). Als het grootste element in een bepaalde ronde al op de juiste plaats staat, moeten er die ronde geen veranderingen plaatsvinden.
 - d. (5) Hoe vaak wordt de functie van **b** minimaal/maximaal aangeroepen tijdens het sorteren van het array `A`? Geef voor beide situaties een voorbeeld met $n = 4$ en $n = 10$.

2. (25 punten)
 - a. (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 ook nog *formeel* en *actueel* parameters. Leg deze zes begrippen duidelijk uit.
 - b. (6) Gegeven een C++-programma met daarin de volgende twee functies:

```
int cor (int u, char c) {
    u -= (c - '0'); cout << u << ", " << c << endl; return u;
} //cor
int don (int x, int y) {
    int i = 9;
    for ( i = 0; i < x; i++ ) { y = y + cor (x,(char)(i + '0')) }; //for
    cout << i << endl; return x + y;
} //don
```

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

```
x = 4; y = 7; cout << don (x,y) << endl; cout << x << ", " << y << endl;
```

- c. (6) Als **b**, maar nu met een `&` (“ampersand”) bij de parameters `u`, `x` en `y` van de functies.
- d. (4) Net als bij **c** voegen we drie `&`'s toe. Wat gaat er mis (geef enkele stappen) bij:

```
x = 10; y = 2; cout << don (x,x) << endl; cout << x << ", " << y << endl;
```

- e. (3) Mag ergens in de functie `cor` staan `u = don (u,-7)`? Onderscheid gevallen met en zonder `&`.

3. (25 punten) Gegeven is een m bij n (beide `const int > 0`; ze hoeven bij deze opgave niet te worden doorgegeven als parameter) array A met integers. Spelers 1 en 2 spelen *vier op 'n rij*. De lege plekken worden aangegeven met 0. De stenen vallen altijd zoveel mogelijk naar beneden. Speler 1 begint. Hiernaast een voorbeeld met $m = 4$ en $n = 5$.

a. (4) Schrijf een C++-functie `int plaats (A,k)` die berekent op welke rij een nieuwe steen in kolom k terecht komt. Als er geen plaats is in kolom k (of als de kolom niet bestaat), geef dan -1 terug. De bovenste rij is rij 0. Voor $k = 4$ is het antwoord 1.

b. (5) Schrijf een Booleaanse C++-functie `boolean beneden (A)` die precies dan `true` teruggeeft als alle stenen in het array A zoveel mogelijk naar beneden zijn gevallen (zoals in het voorbeeld).

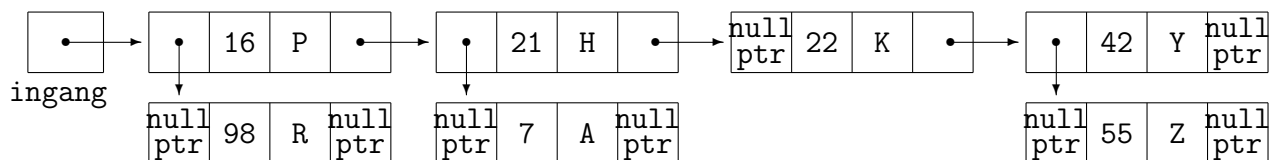
c. (7) Schrijf een C++-functie `int wie (A)` die berekent wie er in het gegeven array aan de beurt is. Als het array A geen geldige spelsituatie is, geef dan 0 terug. Je hoeft niet te controleren of er al “gewonnen” is.

d. (9) Schrijf een Booleaanse C++-functie `boolean vier (A,p,q)` die precies dan `true` teruggeeft als de zojuist in $A[p][q]$ gezette steen *horizontaal* vier op 'n rij maakt: met inbegrip van deze nieuwe staan er daar dan horizontaal minstens vier stenen aaneengesloten van dezelfde kleur. Neem aan dat $0 \leq p < m$ en $0 \leq q < n$.

4. (25 punten) Gegeven is het volgende type:

```
class pers { public: pers* rest; char naam; int lt; pers* volg; };
```

Hiermee wordt een enkel-verbonden lijst met personen opgebouwd, met `naam` en leeftijd (`lt`), gesorteerd op leeftijd. Het veld `rest` bevat een pointer naar een `pers`-object met meer informatie, en `volg` naar het volgende. Een voorbeeld (ingang van type `pers*`):



a. (5) Schrijf een C++-functie `void verwijder (ingang)` die het eerste `pers`-object uit de lijst (met `ingang` van type `pers*` als `ingang`) verwijdert, mits dat er is. Denk dus aan de lege lijst. Verwijder tevens het `pers`-object dat er wellicht onder hangt.

b. (5) Schrijf een C++-functie `void voegtoe (ingang,name,age)` die een nieuw `pers`-object (geheten `name`, leeftijd `age`) vooraan de lijst toevoegt, mits de persoon 18 jaar of ouder is. Het bijbehorende `rest`-object moet deze zelfde waardes krijgen.

c. (6) Schrijf een C++-functie `void wisselen (ingang)` die de informatie uit de twee `rest`-objecten onder de twee eerste objecten verwisselt — indien deze bestaan: $98 \leftrightarrow 7, R \leftrightarrow A$.

d. (3) 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? Mag het, moet het? Leg duidelijk uit.

e. (6) Schrijf een C++-functie `void weggooien (ingang)` die alle `pers`-objecten waarvan de `rest`-pointer `nullptr` is uit de lijst verwijdert. In het voorbeeld wordt alleen het derde object verwijderd.