Tentamen Programmeermethoden
Donderdag 30 maart 2023
13:15–16:15 uur Informatica

Universiteit Leiden
The Netherlands

Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de heading of locaal 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` ≥ 2) gehele getallen.
a. (5) Schrijf een Booleaanse C++-functie `okee (A,n)` die `true` geeft als direct opeenvolgende elementen in array `A` maximaal één van elkaar verschillen (bijvoorbeeld bij 8 9 10 9 8 7 7 8 7 6 7), en anders `false`.
b. (8) Schrijf een Booleaanse C++-functie `mima (A,n,mi,ma)` die bepaalt of `A` precies één minimum en één maximum heeft, en de bijbehorende indices oplevert in `mi` en `ma` (beide -1 als de functie `false` geeft). In het voorbeeld van **a** wordt het `true`, en `mi` $= 9$ en `ma` $= 2$ (vanwege 6 en 10).
c. (8) Schrijf een C++-functie `sorteer (A,n)` die het array `A` oplopend sorteert tussen (en inclusief) minimum en maximum, mits beide uniek zijn, en anders niets doet. Gebruik *bubblesort* (waarbij je moet stoppen als er geen verwisselingen meer waren) en de functie van **b**. In het voorbeeld van **a** geeft dit: 8 9 6 7 7 7 8 8 9 10 7.
d. (4) Hoeveel vergelijkingen tussen array-elementen doet de functie van **c** precies in het slechtste geval, uitgedrukt in de twee indices die `mima` oplevert, en wanneer is dit?

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

```
int mia (int a, int b) {
   int z = 0; while (a > b) {
      a -= b; b = 2 * b; z++; cout << z << "," << a << "," << b << endl; }//while
   b = b - a; return b; }//mia
int dion (int x, int y, int z) {
   while (x < y) {
      x = x + mia (y,z); z++; y--; cout << x << "," << y << "," << z << endl; }//while
   return y + x; return 42; }//dion
```

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

```
a = 9; b = 2; z = 2; cout << dion (b,a,z) << endl;
cout << a << "," << b << "," << z << endl;
```

c. (6) Als **b**, maar nu met een `&` (“ampersand”) bij de vijf parameters van de functies.
d. (4) Net als bij **c** voegen we vijf `&`’s toe. We vervangen `x = x + mia (y,z);` door `y = x - mia (y,x) + 2;` in `dion`. Wat gebeurt er bij:

```
a = 2; b = 3; z = 1; cout << dion (a,b,z) << endl;
```

Zijn er verschillende antwoorden mogelijk, en waarom? Zo ja, geef deze.
e. (3) Mag ergens in de functie `mia` staan `z = dion (z,3 * z,a)`? 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 `P` met kleine letters en hoofdletters (karakters, alle verschillend). Een voorbeeld met `m` $= 4$ en `n` $= 5$ staat hiernaast. Een `'#'` representeert een "leeg" vakje.

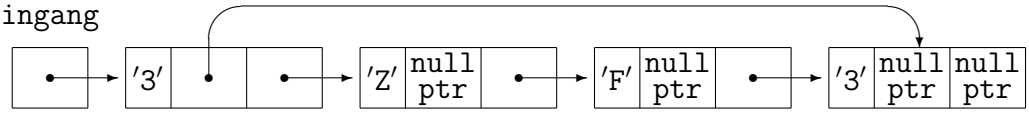
```
u V # I y
a B c d #
A p j e s
# # J E S
```

a. (7) Schrijf een C++-functie `int klinkers (P)` die bepaalt hoeveel klinkers (`'a'`, `'e'`, `'i'`, `'o'`, `'u'`) in `P` staan. Hoofdletters tellen niet. In het voorbeeld: 3.
b. (8) Schrijf een C++-functie `int aantal (P,k)` die telt hoeveel rijen in `P` *minstens* `k` `'#'`'s bevatten. In het voorbeeld, met `k` $= 1$, is dat 3. Stop hierbij in een rij direct met vergelijken zodra `k` `'#'`'s gezien zijn.
c. (10) Schrijf een Booleaanse C++-functie `komtvoor (P,i,j,W,lang)` die bepaalt of het woord `W` (`char W[100];`), met `lang` kleine letters, in `P` voorkomt, door steeds naar direct horizontaal of verticaal aangrenzende posities te gaan. Als dat zo is, moeten `i` en `j` het rij- en kolomnummer worden waar `W` in `P` begint (en anders worden beide -1). In het voorbeeld, met `lang` $= 3$ en woord `'a' 'a' 'p'` in `W`, levert de functie `true` en `i` $= 1$ en `j` $= 0$. We identificeren hierbij kleine letters en hoofdletters. Neem aan dat er geen posities zijn met "dezelfde" buren.

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

```
class persoon { public: char naam; persoon* verder; persoon* volg; };
```

Hiermee wordt een lijst met personen opgebouwd (geheten `naam`). Het veld `verder` bevat een pointer naar het eerstvolgende `persoon`-object met dezelfde naam (`nullptr` als dat er niet is), en `volg` wijst naar het direct volgende `persoon`-object. Een voorbeeld (`ingang` van type `persoon*`):



a. (4) Schrijf een C++-functie `verwijder (ingang)` die het eerste `persoon`-object uit de lijst (met `ingang` van type `persoon*` als ingang) verwijdert, indien dat bestaat en als `naam` een cijfer heeft (zoals in het voorbeeld).
b. (5) Schrijf een C++-functie `wissel (ingang)` die de namen van de twee eerste objecten (mits deze bestaan) verwisselt. Denk aan het goed zetten van de twee `verder`-pointers. Zorg ervoor dat het ook klopt als de namen gelijk zijn.
c. (6) Schrijf een C++-functie `voegtoe (ingang,name)` die een nieuw `persoon`-object (geheten `name`) vooraan de lijst toevoegt. Hierbij moet ook de `verder`-pointer wijzen naar het eerstvolgende `persoon`-object met dezelfde naam.
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. (7) Schrijf een C++-functie `int tel (ingang,name)` die telt hoeveel personen geheten `name` in de lijst staan. Maak hierbij efficiënt gebruik van de `verder`-pointers. In het voorbeeld, met `name` $= '3'$, is het antwoord 2.