Tentamen Programmeermethoden
Woensdag 17 januari 2024
13:00–16:00 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 `char A[n]` staan `n` (een `const int` ≥ 1) hoofdletters en kleine letters.
a. (2) Schrijf een Booleaanse C++-functie `hoofd (kar)` die bepaalt of karakter `kar` een hoofdletter is. Gebruik geen functies als `toupper`.
b. (5) Schrijf een efficiënte Booleaanse C++-functie `gelijk (A,n)` die `true` geeft als ergens in `A` een bij elkaar horende kleine letter en hoofdletter direct naast elkaar staan, en anders `false`. Dus `d e F f g J` levert `true` (wegens `F` $\leftrightarrow$ `f`), en `c a b C B c A` geeft `false`.
c. (10) Schrijf een C++-functie `int lang (A,n,begin,hoeveel)` die de lengte van een langste aaneengesloten serie klinkers (kleine letters `a/e/i/o/u`) teruggeeft. In `begin` moet de beginindex staan van het eerste voorkomen van zo'n serie, in `hoeveel` het aantal keer dat zo'n serie voorkomt (beide -1 als `A` geen kleine klinkers bevat). Het resultaat is 2 voor `A a D a i n e e H o E i u P e e`, `begin` moet 3 worden (voor `a i`) en `hoeveel` 4.
d. (6) Geef een C++-functie `sort (A,n)` die met behulp van *bubblesort* het array `A` van klein naar groot sorteert. Hierbij moeten eerst de kleine letters komen in de juiste volgorde, en dan de hoofdletters. Het tweede array van **b** wordt dus `a b c c A B C`.
e. (2) Hoeveel vergelijkingen tussen array-elementen doet `sort`, uitgedrukt in `n`?

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. (9) Gegeven een C++-programma met daarin de volgende twee functies:

```
int bobbie (int x, int y) { int a = -1;
  while ( y > 1 ) {
    if ( y <= x ) { a = (x - 1) * (y - 1) - 1; }//if
    else { a = y - x; }//else
    cout << a << endl; y--; }//while
  cout << "Bobbie " << a << endl; return a; return 2024; }//bobbie
int ernst (int x, int y) { int i = 7;
  for ( i = x; i < y; i *= 2 ) {
    x = x - bobbie (i,a); }//for
  cout << a << "," << x << "," << i << endl; return x + y; }//ernst
```

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

```
x = 2; y = 5; a = 3; cout << ernst (x,y) << endl;
cout << a << "," << x << "," << y << endl;
```

c. (7) Als **b**, maar nu met een `&` ("ampersand") bij de vier parameters van de functies.
d. (3) Mag ergens in de functie `bobbie` staan `a = ernst (42 * y,x);`? 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 `W` met "hoogtes" (niet-negatieve integers) van een skigebied. Een voorbeeld met $m = 4$ en $n = 5$ staat hiernaast. Hier heeft element `W[2][3]` hoogte 2.

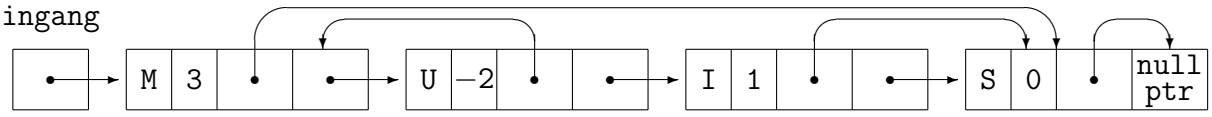
```
4 6 5 1 3
5 9 1 0 1
9 8 9 2 5
7 6 5 4 6
```

a. (8) Schrijf een C++-functie `int top (W)` die het aantal bergtoppen berekent in `W`. Een *bergtop* is een element dat strikt groter is dan al zijn orthogonale (horizontaal/verticaal) buren. Bijvoorbeeld `W[0][4]` is een bergtop. Het voorbeeld heeft 5 bergtoppen.
b. (7) Schrijf een Booleaanse C++-functie `verboden (W,p,q)` die precies dan `true` teruggeeft als (minstens) twee orthogonaal aangrenzende elementen in `W` dezelfde waarde hebben, en anders `false`. Geef in het `true`-geval de coördinaten van één van de betreffende locaties terug via `p` en `q`; anders (zoals in het voorbeeld) moeten beide -1 worden.
c. (10) Schrijf een C++-functie `int route (W,i,j,d)` die berekent in hoeveel stappen je omlaag kunt skieën in `W` als je op positie `(i,j)` begint door steeds een stijlste afdaling (naar één van de orthogonale buren; kies zelf bij gelijk hoogteverschil) te nemen. Neem aan dat $0 \leq$ `i` $<$ `m` en $0 \leq$ `j` $<$ `n`. Bereken ook het gemiddelde hoogteverschil van de route in `d`. In het voorbeeld, voor startpunt $(2, 1)$, heeft de route 5 stappen, via $(3, 1)$, $(3, 2)$, $(3, 3)$, $(2, 3)$ en $(1, 3)$, met gemiddeld hoogteverschil $(8 - 0)/5 = 8/5 = 1.6$.

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

```
class mens { public: char naam; int sprong; mens* ander; mens* volg; };
```

Hiermee wordt een lijst met mensen (geheten `naam`) opgebouwd. Het veld `ander` bevat een pointer naar het `sprong` ≥ -2 verder gelegen `mens`-object, en `volg` wijst naar het volgende `mens`-object. Als `sprong` negatief is, wijst `ander` terug in de lijst; en als `sprong` 0 is, wijst `ander` naar zichzelf. Als de `ander`-pointer voorbij het begin zou wijzen, wijst hij het eerste object aan; analoog bij het laatste. Een voorbeeld (`ingang` van type `mens*`):



a. (6) Schrijf een C++-functie `verwijder (ingang)` die het eerste `mens`-object uit de lijst (met `ingang` van type `mens*` als ingang) verwijdert, mits dat er is. Denk dus aan de lege lijst. Eventuele `ander`-pointers moeten ook goed gezet te worden.
b. (6) Schrijf een C++-functie `voegtoe (ingang,name,ver)` die een nieuw `mens`-object (geheten `name`) vooraan de lijst toevoegt. Hierbij moet `sprong` gelijk worden aan `ver`. Alleen de `ander`-pointer van het nieuwe `mens`-object moet goed worden gezet (en natuurlijk de `volg`-pointer). Neem aan dat $-2 \leq$ `ver` ≤ 1.
c. (3) Schrijf een C++-functie `wisselen (ingang)` die de namen van de eerste en door diens `ander`-pointer aangewezen `mens`-objecten verwisselt — indien deze bestaan, en de `sprong` van de eerste oneven is. In het voorbeeld: `M` $\leftrightarrow$ `S`.
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 aantal (ingang)` die zo snel mogelijk bepaalt hoeveel `mens`-objecten er zijn in de lijst. Gebruik hierbij dus zoveel mogelijk de `ander`-pointers.