

# Tentamen Programmeermethoden

## Donderdag 28 maart 2024

### 13:00–16:00 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  $\geq 3$` ) gehele getallen.
  - a. (4) Schrijf een C++-functie `int verschil (A,n)` die het grootste verschil ( $\geq 0$ ) tussen opeenvolgende waarden uit `A` teruggeeft.
  - b. (5) Schrijf een C++-functie `int bouw (A,X,Y,n)` die array `A` als volgt met getallen vult. Vooraan komen de getallen `X` en `Y`. Vervolgens wordt ieder array-element de som van de twee voorgaande array-elementen. De totale som van de array-elementen moet worden teruggegeven.
  - c. (8) Schrijf een Booleaanse C++-functie `som3 (A,i,j,k,n)` die precies dan `true` teruggeeft als er drie array-elementen in `A` zijn, waarbij één de som van de twee andere is. De parameters `i`, `j` en `k` (met  $i < j < k$ ) moeten de indices worden van deze drie array-elementen ( $-1$  als ze niet bestaan). Als er keuze is, de eerste de beste.
  - d. (5) Geef een C++-functie `sort (A,n)` die met behulp van *bubblesort* het array `A` aflopend sorteert. Stop zodra er geen verwisselingen zijn.
  - e. (3) Hoeveel vergelijkingen tussen array-elementen doet `sort` maximaal? En wanneer gebeurt dat?

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

```
int caro (int a, int b) {
    int x = 42, r = 42; z++; x++;
    for ( r = 1; r <= a; r++ ) { b += a; } //for
    cout << a << ", " << b << ", " << x << ", " << r << ", " << z << endl;
    return a + b; cout << "Nee!" << endl; } //caro
int piet (int a, int b) {
    int x = 42, r = 42; z++; r++; a = caro (a,b) + caro (b,a);
    cout << "Caro " << a << ", " << b << ", " << x << ", " << r << ", " << z << endl;
    return a + b; cout << "Ja!" << endl; } //piet
```

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

```
x = 1; y = 3; z = 10; cout << piet (x,y) << endl;
cout << x << ", " << y << ", " << z << endl;
```

- c. (3) Wat is de returnwaarde van `caro (a,b)` uitgedrukt in de originele `a` en `b`?
- d. (6) Als `b`, maar nu met een `&` (“ampersand”) bij de vier parameters van de functies. Zijn er in principe verschillende antwoorden mogelijk, en waarom?
- e. (3) Mag ergens in de functie `caro` staan `b = piet (x,caro (7,r))`? Onderscheid gevallen met en zonder `&`.

**3.** (25 punten) Gegeven is een  $m$  bij  $n$  (beide `const int > 0`; ze hoeven 4 X K 5 8  
 bij deze opgave niet te worden doorgegeven als parameter) array  $H$  met B B B 4 G  
 karakters: cijfers of hoofdletters. Een voorbeeld met  $m = 4$  en  $n = 5$  V V 4 9 3  
 staat hiernaast. Hier heeft element  $H[2][3]$  waarde '9'. Y K Z 2 Y

**a.** (5) Schrijf een Booleaanse C++-functie `cijfers` ( $H$ ) die bepaalt of het aantal cijfers in  $H$  groter dan of gelijk is aan het aantal hoofdletters. In het voorbeeld `false` ( $8 < 12$ ).

**b.** (10) Schrijf een C++-functie `int cijferkolom` ( $H, \text{som}$ ) die de index van de eerste kolom van  $H$  geeft die geheel uit cijfers bestaat, waarbij `som` hun som moet gaan bevatten. Als zo'n kolom er niet is  $-1$  (en `som` moet  $0$  worden). In het voorbeeld  $3$ , met `som` gelijk aan  $5 + 4 + 9 + 2 = 20$ . Denk aan het converteren van karakters naar getallen.

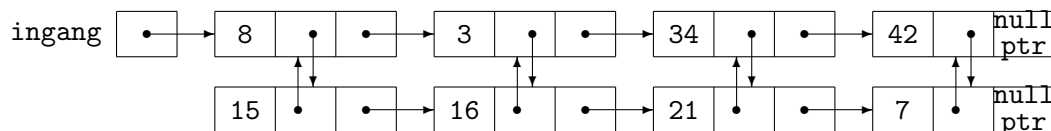
**c.** (10) Stel dat je begint in positie  $(i, j)$  met  $0 \leq i < m$  en  $0 \leq j < n$ . Als het betreffende karakter een even getal voorstelt ga je naar rechts, bij een oneven getal naar beneden. Je herhaalt dit tot je niet meer verder kunt omdat je of het array bent uitgelopen of omdat je in een hoofdletter bent beland, en telt het aantal gedane stappen. Vanuit  $(2, 2)$  kun je zo  $3$  stappen doen ( $4 \rightarrow 9 \rightarrow 2 \rightarrow Y$ ); vanuit  $(3, 1)$   $0$  stappen; vanuit  $(0, 4)$   $1$  stap.

Schrijf een Booleaanse C++-functie `wandel` ( $H, u, v, d$ ) die precies dan `true` teruggeeft als je ergens in  $H$  beginnend minstens  $d$  stappen kunt doen, en zo ja een dergelijke positie in  $(u, v)$  oplevert (anders moeten beide  $-1$  worden). Met  $d = 3$ : `true`,  $u = v = 2$ .

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

```
class vakje { public: int info; vakje* onder; vakje* rechts; };
```

Hiermee wordt een structuur met twee rijen met vakjes opgebouwd, bestaande uit een getal, en twee pointers; zie onderstaand plaatje. Voor beide rijen geldt: de `rechts`-pointer wijst naar het volgende vakje (als laatste een `nullptr`). In de bovenste rij wijst de `onder`-pointer naar het onder gelegen vakje, in de onderste rij terug naar boven. Een voorbeeld, met `ingang` van het type `vakje*`:



**a.** (5) Schrijf een C++-functie `verwijder` (`ingang`) die het eerste tweetal vakjes uit de lijst (met `ingang` als `ingang`) netjes verwijdert, indien deze bestaan. In het voorbeeld die met  $8$  en  $15$ .

**b.** (6) Schrijf een C++-functie `voegtoe` (`ingang, x, y`) die twee nieuwe vakjes onder elkaar, met  $x$  en  $y$  erin vooraan de lijst met `ingang` toevoegt.

**c.** (5) Schrijf een C++-functie `desom` (`ingang`) die de waarde van het eerste vakje (als dat er is) ophoogt met de waarde in het vakje daar direct rechtsonder (indien dat bestaat), mits hun som deelbaar is door  $4$ . In het voorbeeld:  $8 \rightarrow 8 + 16 = 24$ .

**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 `wissel` (`ingang`) die voor eerste ( $8 \leftrightarrow 15$ ), derde ( $34 \leftrightarrow 21$ ), vijfde, ... "oneven" tweetallen in de lijst de getallen (de `info`-velden) uit het bovenste en het onderste vakje met elkaar verwisselt.