
Dit document bevat opgaven voor zelfstudie bij het eerstejaars college *Programmeermethoden*, Universiteit Leiden, najaar 2010, zie (ook voor de antwoorden)

www.liacs.nl/home/kosters/pm/

Met dank aan allen die aan deze tekst hebben bijgedragen.
Walter A. Kosters, Leiden, 23 augustus 2010.

Opgaven

1. Geef type en zo mogelijk de waarde van de volgende uitdrukkingen. Hierbij zijn p , q , r en s variabelen van type `bool`, en k van type `int`.

a. `sqrt (4)` b. `sqrt (4.0)` c. `'t' - 'e'`
d. `ceil (-99.9)` e. `- floor (99.9)` f. `- floor (-99.9)`
g. `!(p && q) == !(!p && !q)`
h. `10 % 3` i. `10 / 3` j. `126 / 3 % 5`
k. `(p && (q && !q)) || !(r || (s || !s))`
l. `(floor (-65.3) < fabs (-65.3)) && p`
m. `odd (k) || odd (k+1)`

2. a. Schrijf een C++-programma dat de gebruiker om een temperatuur in graden Fahrenheit als invoer vraagt. Na het inlezen hiervan wordt deze temperatuur omgerekend in graden Celsius, waarna beide getallen afgedrukt worden. Gebruik: $\text{Temp}(\text{in Celsius}) = (5/9) * (\text{Temp}(\text{in Fahrenheit}) - 32)$.
- b. Idem, maar nu mag alleen met gehele getallen gerekend worden. In dit geval moet er dus afgerond worden.
3. Schrijf een C++-programma dat het aantal seconden vanaf middernacht aan de gebruiker vraagt en de tijd uitvoert als: 'uren:minuten:seconden'. Probeer ook eens uit zoeken hoe je de tijd uit het computersysteem kunt aflezen via C++.
4. Wat is het grootste gehele getal dat in C++ in een `int` past? Waarom dit getal? Hangt dit af van de C++-implementatie? En hoe staat het bij een `long`? Wat gebeurt er als je 1 optelt bij het grootste te representeren getal?
Wat is het grootste getal dat in een `double` past? En het kleinste positieve (groter dan nul) getal?
5. Geef kritiek op de het volgende stukje C++, waarbij de variabele doorgaan van het type `bool` is: `if (doorgaan = true) ...`
6. a. Laat zien dat elk `do-while`-statement herschreven kan worden met behulp van `if`- en `while`-statements.
- b. Laat zien dat elk `while`-statement herschreven kan worden met behulp van `if`- en `do-while`-statements. Kan het ook zonder `if`-statements?
7. Een slecht geschreven C++-programma bevatte het volgende statement:

```

if ( a < b ) if ( c < d ) x = 1;
else if ( a < c ) if ( b < d ) x = 2;
                else x = 3;
else if ( a < d ) if ( b < c ) x = 4;
                else x = 5;
                else x = 6;
                else x = 7;

```

- a. Herschrijf dit statement; gebruik een betere layout.
 - b. Staan er overbodige of tegenstrijdige condities in?
 - c. Schrijf een eenvoudiger statement dat hetzelfde effect heeft.
8. Schrijf een programma dat drie gehele getallen inleest en deze in volgorde van grootte afdruckt. Geef enkele verschillende methoden. Zo kunnen de getallen alleen op het beeldscherm gesorteerd afgedrukt worden, maar ook intern in het programma gesorteerd worden.
9. Schrijf een programma dat een geheel getal n inleest, en vervolgens (als tenminste $n \geq 0$) $n! = n * (n - 1) * \dots * 2 * 1$ berekent. Als $n < 0$ moet er een passende foutmelding afgeleverd worden. Gebruik hierbij:
- a. het while-statement,
 - b. het for-statement,
 - c. het do-while-statement.
- Welke mogelijkheid levert het “beste” programma?
Schrijf ook een functie (zonder recursie) die $n!$ berekent.
10. Schrijf een functie `spaties (int aantal)` die `aantal` spaties op het beeldscherm zet.
11. Omschrijf een klasse-structuur voor de volgende situaties:
- a. een paar reële coördinaten met een plaatsnaam,
 - b. een studentenadministratie op een faculteitsburo,
 - c. idem, voor het sportcentrum.
- Stel dat voor de situatie van b een (groot) programma geschreven moet worden om de studentenadministratie bij te houden. Geef de benodigde functie-headings, met commentaar.
12. Schrijf een programma dat als invoer om een rij positieve getallen vraagt, afgesloten door een getal kleiner dan of gelijk aan nul. Vervolgens moeten het maximum, het minimum en het gemiddelde van deze getallen (het laatste getal niet inbegrepen) bepaald worden. Is het for-statement of het while-statement beter geschikt?
13. Schrijf een programma dat voor elk bedrag kleiner dan een euro het kleinste aantal munten bepaalt dat samen dit bedrag oplevert. Gebruikt mogen worden: munten van één, twee, vijf, tien, twintig en vijftig eurocent. De gebruikte munten moeten ook worden afgedrukt.

14. Schrijf een programma dat uitdrukkingen als $56-664-77+2-888$; inleest en berekent. Elk (geheel) getal wordt voorafgegaan door een teken, terwijl de hele uitdrukking door een punt-komma wordt afgesloten. Schrijf ook een versie die echt karakter voor karakter inleest (met `kar = cin.get ();`).

15. Schrijf een programma dat voor gegeven gehele n het volgende berekent:

$$A(n) = 1/2 + 2/4 + 3/8 + 4/16 + \dots + n/2^n.$$

Gebruik een functie; maak hierbij *geen* aparte functie voor machtsverheffen. Maakt het overigens uit in welke volgorde de sommatie berekend wordt?

16. Voorspel de uitvoer van de volgende twee programma's:

<pre>#include <iostream> using namespace std; int iets; void wat (int& a, int& b) { a = -1; b = -2 * a; cout << a << b << endl; } // wat int main () { iets = 1; wat (iets,iets); cout << iets << endl; return 0; } // main</pre>	<pre>#include <iostream> using namespace std; int iets; void bla (int &a, int & b) { a = 10 * b + iets / 2; cout << a << b << endl; } // bla int main () { iets = 10; bla (iets,iets); cout << iets << endl; return 0; } // main</pre>
--	---

Wat gebeurt er als `&` weggelaten wordt in de headings?

17. Gegeven is de functie:

```
void test (int x, int& y) {
    int z = 9;
    x = 5;
    y = 6;
    z = 7;
} // test
```

Stel dat x 1 is, y 2 en z 3 voor globale variabelen x , y en z (van type `int`). Wat is de uitvoer geproduceerd door:

- a. `test (x,y); cout << x << y << z << endl;`
- b. `test (y,x); cout << x << y << z << endl;`
- c. `test (1,z); cout << x << y << z << endl;`
- d. `test (z,1); cout << x << y << z << endl;`
- e. `test (z,x); cout << x << y << z << endl;`

18. (Tentamen 6 januari 1997) Gegeven zijn de volgende functies:

```
int f (int x, int y) {
    x--; return x*y; } // f
int g (int a, int b) {
    int x = 3; b += x; a--; a = f (a,b) + f (a,a);
    cout << x << a << b; return a+x-2; } // g
```

a. Neem aan dat de waarden van de *globale variabelen* x en y, beide int, bij binnenkomst van g 6 respectievelijk 16 zijn.

Wat gebeurt er bij `cout << g (x,y) << x << y << endl;`? Wat wordt er afgedrukt? Probeer duidelijke uitleg te geven.

b. Geef een functie `int G (int a, int b)` die dezelfde return-waarde oplevert als g voor alle mogelijke waarden van de parameters, maar die uit slechts één return-statement bestaat, en waarin f niet meer wordt aangeroepen.

c. We voegen vier maal een & toe, en wel bij alle parameters in de headings van f en g. Beantwoord opnieuw vraag a. Geef alle mogelijke uitvoeren, en leg uit waarom er verschillende antwoorden mogelijk zijn.

19. (Tentamen 5 januari 1998) Gegeven zijn de volgende functies:

```
int peter (int r, int s) {
    s--; return r+s+2; } // peter
int ellen (int p, int q) {
    int a = 7; p++; q -= 2;
    for ( a = 2; a < q; a++ ) p = p + peter (p,q);
    cout << a << p << q << endl; return a+p+q; } // ellen
```

a. Neem aan dat de waarden van de *globale variabelen* a en b, beide int, bij binnenkomst van de functie ellen 2 respectievelijk 6 zijn.

Wat gebeurt er bij `cout << ellen (a,b); cout << a << b << endl;`? Wat zou er worden afgedrukt? Probeer duidelijke uitleg te geven.

b. We voegen vier maal een & toe, en wel bij alle parameters in de headings van ellen en peter. Beantwoord opnieuw de vorige vraag.

c. Vervang in de functie ellen het statement `p = p + peter (p,q);` door het statement `p = p + peter (q,p);`. Zet de vier &'s er weer bij, net als bij b. Leg uit waarom uiteindelijk de globale variabele a verschillende waarden kan hebben, en geef deze.

20. Schrijf een functie die x tot de macht y berekent, waarbij

a. x en y beide van type double zijn,

b. x van type double is, y van type int, en exp en log (uit math.h of cmath) niet gebruikt mogen worden.

Gebruik overloading. Vergelijk het resultaat eens met pow uit math.h of cmath.

21. Gegeven een functie F:

```
int F (int& x) {
    x = x + 4;
    return x;
} // F
```

- Wat gebeurt er bij `cout << (x + F(x));` (`x` is een globale variabele van type `int` met waarde 7)?
- Wat gebeurt er bij `cout << (F(x) + x);` ?
- Idem, als `&` weggelaten wordt.
- Is het verstandig als functies die een `int` als resultaat hebben van “call by reference” gebruik maken?

22. Schrijf een functie `drukaf (char letter)` die de letters a, b en c afdruckt als:

```
****      ****      ****
*  *      *  *      *
****      ,  ****      ,  *
*  *      *  *      *
*  *      ****      ****
```

23. Schrijf een functie `Pyramide (int hoogte)` die een pyramide, bestaande uit sterretjes, en ter hoogte `hoogte` afdruckt (zie beneden voor `hoogte = 4`). Er moet getest worden of `hoogte` positief is. Als `hoogte` negatief is gebeurt er niets.

```
  *
 ***
*****
*****
```

- Schrijf een functie die elk voorkomen van het woordje "er" in een (onzin)tekst verandert in "ER"; tevens moet het aantal regels van de tekst bepaald worden.
- Schrijf een functie die voor elke ingevoerde datum tussen 1 januari 1900 en 31 december 2100 de bijbehorende dag afdruckt. Gegeven is dat 1 januari 2002 op een dinsdag viel. Het jaar 2000 is overigens wel een schrikkeljaar, maar 1900 en 2100 niet.
- Schrijf een functie `fibonacci (int n)` die het `n`-de getal van Fibonacci `Fibonacci (n)` berekent. Er geldt `Fibonacci (1) = Fibonacci (2) = 1` en `Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2)`, voor `n > 2` (gebruik nog geen recursie, zie later).
- Schrijf een functie die waarden van de zogenaamde Hermite-polynomen uitrekenet (zie ook later voor een recursieve versie). Er geldt:
 $H_0(x) = 1, H_1(x) = 2x$ en
 $H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x)$ voor $n > 1$.
- In een zeker C++-programma moeten de functies P, Q en R voorkomen. In de functie P wordt gebruik gemaakt van de functies Q en R. Geef zoveel mogelijk verschillende manieren om dit te programmeren.

29. Het te betalen bedrag voor het sturen van een pakketje overzee wordt als volgt berekend (nemen we aan). Eerst wordt het gewicht naar boven afgerond op het dichtstbijzijnde veelvoud van 15 gram. Dan wordt het tarief uit de volgende tabel afgelezen:
- Als het gewicht 15 gram is, wordt het tarief 120 eurocent,
 - Als het gewicht 30 gram is, wordt het tarief 220 eurocent,
 - Als het gewicht 45 gram is, wordt het tarief 310 eurocent,
 - Als het gewicht 60 gram is, wordt het tarief 360 eurocent, plus 20 eurocent per volledige 1000 km,
 - Als het gewicht 75 gram of meer is, wordt het tarief 400 eurocent, plus 30 eurocent per volledige 1000 km.
- a. Maak een switch-statement dat bij gegeven gewicht en —indien nodig— de afstand het tarief uitrekent. Maak hier ook een functie van.
- b. De invoer bestaat nu uit een aantal regels. Elke regel bevat de gegevens van een te versturen pakketje: het gewicht (in grammen) en daarna de af te leggen afstand (in kilometers). De laatste regel bevat alleen het getal 0. Schrijf een programma dat de verzendkosten van al de pakketjes samen berekent. Bovendien moet worden bijgehouden hoeveel pakketjes uit elke gewichtscategorie verstuurd worden.
30. a. Maak een klasse `complexgetal` die een complex getal moet representeren.
- b. Schrijf member-functies voor het afdrukken en het conjugeren van een complex getal.
- c. Schrijf een member-functie die twee variabelen van type `complexgetal` op de juiste manier bij elkaar optelt. Het resultaat komt in de variabele van de klasse waar deze functie behoort.
- d. Idem voor vermenigvuldigen.
31. a. Verbeter de functie `drukaf` uit het programma met de vaas en de knikkers zodat verzamelingen netter afgedrukt worden, bijvoorbeeld als `{2, 4, 7, 16}`. Denk ook aan `∅`, de lege verzameling.
- b. Verbeter het programma zodat de uitvoer “leesbaar” wordt: afgedrukt moeten worden rijtjes als `3 5, 3 9, ...`
32. Schrijf een programma dat een `int array[100]` vult met de eerste honderd kwadraten. Maak hier ook een functie van. Druk het array tevens af.
33. Er is een enquête gehouden onder honderd mensen. De enquête bestond uit drie vragen. Op elke vraag waren slechts de antwoorden ja en nee mogelijk. De resultaten staan in een `array bool uitslag[100][3]`. Zo bevat `uitslag[30][1]` het antwoord van persoon 31 op vraag 2, waarbij `true` ja betekent en `false` nee. Schrijf een functie die voor elk van de acht mogelijke antwoord-combinaties bepaalt bij hoeveel mensen deze voorkomt.
34. Gegeven: `char boek[99][35][60];`
- a. Hoe kun je verwijzen naar de derde letter van de vierde regel op bladzijde 55 van een “boek”?
- b. Kopieer van iedere bladzijde de eerste vijf regels naar een nieuw array.
- c. Verwissel eerste en laatste bladzijde.

35. Geef een programma dat een tekst regel voor regel inleest en elke regel van achter naar voren afdrukt. Hierbij moet een `char regel[80]` gebruikt worden om de regel in op te slaan, aannemend dat er hoogstens 80 (of 79) tekens per regel staan. Wat verandert er als `char*` gebruikt moet worden?
36. Schrijf een programma dat alle in een invoertekst voorkomende woorden in een array opslaat. Als woorden twee keer (of nog vaker) voorkomen moeten ze maar één keer opgeslagen worden.
37. Schrijf een functie vermenigvuldig (`int A[][n]`, `int v[]`, `int res[]`) die het product van de matrix `A` en de vector `v` in de vector `res` stopt. Neem aan dat `A` een `n` bij `n` matrix is en `v` een vector met `n` componenten.
38.
 - a. Bedenk een array-representatie voor een stand op een schaakbord.
 - b. Is er een betere (minder geheugenruimte) denkbaar?
 - c. Idem voor een stand op een dambord.
39.
 - a. Schrijf een functie die de array-indices oplevert van het grootste en van het op een na grootste getal van een gegeven array `int A[max]`, met $\text{max} \geq 2$.
 - b. Geef in woorden een methode aan die ditzelfde bewerkstelligt, maar dan met minder vergelijkingen tussen gehele getallen. Hint: hoe vind je de twee echt beste spelers van een tennistoernooi?
40. Gegeven `class grootgetal {public: int inhoud[MAX]};`. Hierbij is `MAX` een zekere constante, zeg 25. In een variabele van klasse `grootgetal` wordt een geheel getal met veel cijfers opgeslagen. Schrijf nu een functie die de som van twee variabelen van klasse `grootgetal` uitrekent. Geef methoden om de "lengte" van het getal goed bij te houden.
41.
 - a. Schrijf een functie `omzetten` die getallen als `long` gegeven, omzet in array-vorm (elk array-element bevat een of meer cijfers, zie de vorige opgave).
 - b. Schrijf een functie `terug` die het omgekeerde bewerkstelligt.
 - c. Hoe kun je controleren of voor het getal 1234 de zaak klopt?
42. Gegeven zijn de volgende declaraties in C++:

```
char woord[m];
char verhaal[n];
```

met `m` en `n` constanten (gehele getallen) met $n > m > 0$. Schrijf nu een C++-functie die vertelt of een variabele `woord` voorkomt in een variabele `verhaal`. Als dat zo is moet de array-index die het eerste optreden van `woord` aanduidt afgedrukt worden.

Enkele voorbeelden (in string-notatie):

- `woord = "la"` (`m = 2`), terwijl `verhaal = "leuter babbel bla bla"` (met `n = 21`), met als uitvoer: Ja, `index = 15`.
- `woord = "tja"` (`m = 3`), `verhaal` als boven, levert als uitvoer: Nee.

43. Gegeven een array `bord`: `char bord[8][8]`; . Dit is dus een “schaakbord” gevuld met letters.
- Schrijf een boolese functie in C++ die `true` oplevert als er in `bord` minstens twee aangrenzende (naast elkaar of boven elkaar gelegen) vakjes zijn met dezelfde letter, en anders `false`.
 - Idem, maar nu moet er `true` opgeleverd worden als er alleen hoofdletters in het array zitten opgeslagen.
 - Neem aan dat er alleen hoofdletters worden opgeslagen. Schrijf een functie die bepaalt of er ten minste één rij in `bord` is die geheel uit klinkers ('A', 'E', 'I', 'O', 'U') bestaat. De functiewaarde moet het nummer van zo'n rij zijn en `-1` als zo'n rij niet bestaat.
 - Herschrijf de functie van `c` zo dat er geen overbodige tests uitgevoerd worden. Bijvoorbeeld: als er een rij met alleen klinkers gevonden is hoeven de overige rijen niet meer bekeken te worden.

44. (Tentamen 6 januari 1997) Gegeven zijn:

```
const int m = 30; const int n = 40;
char puzzel[m][n]; int nummers[m][n];
```

In een array `puzzel` zit een kruiswoordraadsel opgeslagen. De zwarte vakjes worden met '#' aangegeven, woorden staan in hoofdletters. Een voorbeeld ($m = 3$, $n = 4$):

H E T #	1 0 2 0
A # O M	0 0 3 4
# S P A	0 5 0 0

- Schrijf een boolese C++-functie `BestaatHorWoord` (`puzzel`, `i`, `j`) die precies dan `true` oplevert als op positie (`i`, `j`), dus beginnend in de `i`-de rij en de `j`-de kolom, een horizontaal woord begint. Het woord moet meer dan één letter hebben, en echt op deze plek beginnen. Aangenomen mag worden dat (`i`, `j`) binnen de puzzel valt.
 - Schrijf een C++-functie `Nummeren` (`puzzel`, `nummers`) die de vakjes van `nummers` nummert zoals de puzzel uit `puzzel` aangeeft (zie voorbeeld). Vakjes waar geen woord begint, bijvoorbeeld de met een '#' gemarkeerde, krijgen een 0. Vakjes waar wel een horizontaal en/of verticaal woord begint, krijgen regel voor regel, van links naar rechts, een rangnummer, beginnend bij 1. Gebruik de functie van `a`, en een soortgelijke functie voor verticale woorden (deze hoeft niet meer geschreven te worden).
 - Schrijf een C++-functie `Woord` (`puzzel`, `w`) die het `w`-de woord uit `puzzel` afdrukt, als hier een horizontaal woord staat, en anders niets. Neem aan dat de puzzel minimaal `w` woorden heeft. In het voorbeeld wordt bij 1 HET afgedrukt, bij 2 niets en bij 3 OM.
45. (Tentamen 5 januari 1999) In een array `T` (`int T[m][n]` met `m` en `n` constanten) wordt van `m` personen bijgehouden wie op welke tijdschriften geabonneerd is. Zo betekent `T[3][2] == 13` dat klant 3 tijdschrift 13 leest; de 2 heeft geen speciale betekenis. Als een array-element echter 0 is, wordt het nog niet, of niet meer, gebruikt voor een tijdschrift—de persoon in kwestie heeft dan dus minder dan `n` abonnementen, het maximale aantal.

Per klant zijn de tijdschriftnummers verschillend en oplopend gesorteerd, maar er kunnen nullen tussen de niet-nullen zitten.

Twee voorbeelden die dezelfde situatie beschrijven (met $m == 3$ en $n == 4$):

0	3	10	13	3	10	13	0
0	0	10	0	10	0	0	0
1	5	0	10	1	5	10	0

- a. Schrijf een `int C++`-functie die de persoon met de meeste abonnementen (de index van de rij met de meeste niet-nullen) geeft. Als er meer dan één persoon is die het grootste aantal abonnementen heeft, doet het er niet toe welke wordt opgeleverd. In het voorbeeld hierboven: 0 of 2.
 - b. Schrijf een `void C++`-functie die in iedere rij de niet-nullen zoveel mogelijk naar links aanschuift—van het voorbeeld linksboven naar dat rechtsboven.
 - c. Schrijf een `int C++`-functie die oplevert hoeveel tijdschriften door meer dan één persoon gelezen worden. In het voorbeeld zou het antwoord 1 zijn. Als het helpt, mag aangenomen worden dat de tijdschriftnummers tussen 1 en 999 liggen, grenzen inbegrepen.
46. a. Bedenk zelf methoden om willekeurige (“random”) getallen te genereren. Geef voor- en nadelen.
- b. Bekijk de lineaire congruentie methode. Bereken $y = (3 * x + 5) \% 16$; , waarbij x om te beginnen 1 is, dus y wordt 8: het eerste random-getal. Vervang vervolgens x door y en bereken de nieuwe y ; deze is 13: het tweede random-getal. Ga zo door. Reken de rij verder uit. Wat zijn (lijken) voor- en nadelen van deze methode?
- c. Probeer de getallen van b zo te kiezen dat de methode beter loopt.
47. Schrijf een recursieve functie die x tot de macht y berekent, waarbij y een natuurlijk getal is (zie ook vroeger). Probeer dit op twee verschillende manieren te doen. Zeg iets over het verschil in complexiteit.
48. Schrijf een recursieve functie die
- a. het n -de Fibonacci-getal berekent (zie ook vroeger),
 - b. waarden van het n -de Hermite-polynoom berekent (zie ook vroeger).
49. Gegeven een array `int A[30]`. Schrijf een recursieve functie die `true` oplevert precies dan als een gegeven getal in het array voorkomt.
50. We hebben de volgende drie functies:

```
void keerom1 ( ) {
    char kar;
    cin >> kar;
    if ( kar != '.' )
        keerom1 ( );
    cout << kar;
} // keerom1

void keerom2 (char kar) {
    if ( kar != '.' ) {
        cin >> kar;
        keerom2 (kar);
    } // if
    cout << kar;
} // keerom2
```

De functie `keerom3` is als `keerom2`, maar met elke 2 vervangen door 3 en als functiehoofd `keerom3(char& kar);`.

De invoertekst is: "De parterretrap was vies.". Stel dat `kar` een globale variabele is van type `char`. Wat gebeurt er

- a. bij `keerom1 ();`,
- b. bij `cin >> kar; keerom2 (kar);`,
- c. bij `cin >> kar; keerom3 (kar);?`

51. Het gegevensbestand van een schilderijenverzameling wordt opgeslagen in een array van schilderijen: `schilderij bestand[max]`, waarbij `schilderij` een klasse is van een bepaalde vorm. Voor elk schilderij worden onder meer opgeslagen: catalogusnummer, naam van de maker (indien bekend) en jaar waarin het gemaakt is (idem).

- a. Beschrijf de klasse `schilderij`.
- b. Schrijf een functie die de gegevens afdruckt van alle schilderijen die voor 1800 gemaakt zijn.
- c. Schrijf een functie die een zeker catalogusnummer opzoekt, melding geeft van het eventueel niet voorkomen, en anders vraagt of de gegevens veranderd moeten worden.
- d. Schrijf een functie voor het creëren van een index-array (ook wel indexbestand genoemd), waarmee het bestand in alfabetische volgorde van schildersnaam doorlopen kan worden.
- e. Idem, maar nu voor de jaartallen.
- f. Geef aan hoe een en ander werkt voor het volgende bestand(je):

array-index	cat-nr	naam	jaar
0	546	Matisse	1934
1	12	Appel	1977
2	111	Rembrandt	1654
3	554	?	1666
4	33	Matisse	1912
5	274	Mondriaan	1934
6	849	Rembrandt	?

52. Lineair zoeken is een voor de hand liggende methode om een getal in een array op te zoeken. In een reeds gesorteerd array kan een getal snel worden opgespoord met binair zoeken.

- a. Gegeven de rij 1, 3, 9, 10, 13, 17, 19, 21, 28. Hoe verloopt het zoeken naar het getal 9? En naar 21? En naar 18? Beantwoord de vraag zowel voor lineair als voor binair zoeken.
- b. Stel dat het aantal elementen in de rij $2^k - 1$ (met $k > 0$ geheel) is, bijvoorbeeld $8 - 1 = 7$ of $16 - 1 = 15$. Hoeveel vergelijkingen heb je nodig om het eerste array-element te vinden? En hoeveel om te constateren dat een bepaald element niet voorkomt? Beantwoord de vraag wederom zowel voor lineair als voor binair zoeken.

53. a. Pas de sorteermethode bubblesort zo aan dat er gestopt wordt zodra er een ronde zonder verwisselingen geweest is. Pas de methode vervolgens zo aan dat bij iedere ronde begonnen wordt bij het eerste element dat wellicht verkeerd zou kunnen staan, en geëindigd wordt bij het laatste element dat wellicht verkeerd zou kunnen staan (op grond van kennis uit de vorige ronde).
- b. Hoeveel vergelijkingen van array-elementen doet de oorspronkelijke bubblesort, respectievelijk de aangepaste versie, bij het sorteren van n getallen in het beste geval en in het slechtste geval?
54. (Tentamen 4 januari 1996) Gegeven zijn `const int n = 1000;` en `int A[n];`. De variabele `A` bevat een rij onderling verschillende getallen.
- a. Schrijf nu een C++-functie `bergop (A, i, j)` die het array-element `A[i]` (met i tussen 0 en $j-2$) op de juiste plaats opbergt tussen de reeds oplopend gesorteerde array-elementen `A[i+1], A[i+2], ..., A[j-1]`.
- b. Schrijf een C++-functie `sorteer (A)` die het array `A` oplopend sorteert door de functie `bergop` herhaald aan te roepen.
- c. Hoeveel vergelijkingen tussen array-elementen doet dit sorteer-algoritme voor het rijtje $n, n-1, \dots, 1$?
- d. Is de methode beter dan, gelijkwaardig met of slechter dan de gewone bubblesort qua complexiteit (gedaan aantal vergelijkingen), zowel in het beste als in het slechtste geval?
- e. Geef C++-code voor `invoegsorteer`.
55. De sorteermethode Shellsort moet gebruikt worden om de volgende rij in volgorde van grootte te sorteren: 503, 087, 512, 061, 908, 170, 897, 275, 653, 426, 154, 509, 612, 677, 765, 703.
- a. We doen dit als volgt: we zetten eerst de getallen (op positie) 1 en (op positie) 9 goed ten opzichte van elkaar (503 en 653), dan getallen 2 en 10, enzovoorts. De stapgrootte is dus 8, en de rij wordt “8-gesorteerd”. Zet vervolgens de getallen 1, 5, 9 en 13 goed, enzovoorts. De stapgrootte is nu 4. Vervolgens alle elementen die 2 van elkaar af staan en tenslotte de hele rij. Het sorteren van de deelrijen gaat met een of andere “snelle” methode.
- b. Bij a spelen de getallen 8, 4, 2 en 1 een rol. Sorteert de rij nu door het rijtje 7, 5, 3 en 1 te gebruiken.
- c. Geef een wat formelere beschrijving van deze sorteermethode.
- d. Sorteert de rij ook met bubblesort, invoegsorteer en functie `simpelsort`. Tel de aantallen vergelijkingen tussen array-elementen die de verschillende methoden nodig hebben.
56. a. Sorteert met behulp van quicksort de rij 3, 7, 2, 5, 1, 9, 4.
- b. Wat doet quicksort met een al gesorteerde rij?
- c. Wat doet quicksort met een omgekeerd gesorteerde rij zoals 4, 3, 2, 1?
57. Gegeven:

```

int getal, cijfer;
int* p, *q;
int* r, s;
int A[10];

```

- a. Wat stellen p, *p, &p, q, s, &getal, A, A[3] en A[10] voor?
- b. Geef equivalente expressies voor A[3].
- c. Is int* p, q; hetzelfde als int* p; int* q; of als int *p, *q;?
- d. Wat betekenen A[9], *A[9], &A[9], (*A)[9] en *(A[9])?

Stel dat getal de waarde 4 heeft en cijfer de waarde 7. Wat gebeurt er in de volgende gevallen:

- e. p = getal;
- f. *q = cijfer;
- g. p = new int; p = cijfer;
- h. q = new int; *q = getal;
- i. p = new int; p = NULL; (is dit overigens in het algemeen handig?)

Stel dat p wijst naar een int met waarde 16 en q naar een int met waarde 75.

- j. Verwissel deze waarden met behulp van de variabele getal. Geef verschillende manieren om dit te bewerkstelligen.
- k. Laat beide naar het getal dat in cijfer staat wijzen.
- l. Stel dat $A[i] = i$ voor $i = 0, 1, 2, \dots, 9$. Laat de pointer r naar een zelfde array als A verwijzen. Laat q hetzelfde aanwijzen.
- m. Zorg er vervolgens voor dat q een array “is” waarin de eerste 10 kwadraten zijn opgeslagen. Verandert het door r?

58. Bekijk het volgende programma:

```

#include <iostream>
using namespace std;

void tjatja (int* & r, int* & s) {
    r = new int;
    *r = 1;
    *s = 96;
} // tjatja

int main ( ) {
    int* p; int* q;
    p = new int;
    *p = 3;
    q = new int;
    *q = 4;
    cout << *p << *q << endl;
}

```

```

    tjatja (p,q);
    cout << *p << *q << endl;
    return 0;
} // main

```

- Wat stellen `p`, `*p` en `&p` voor?
- Wat zijn hier actuele en formele parameters?
- Wat is de uitvoer van dit programma (met uitleg/plaatjes)?
- Idem als in de functie-heading van `tjatja` het symbool `&` twee maal wordt weggelaten.
- Pas de programma's van c en d zo aan (met `delete`) dat na afloop er geen vakjes meer in het geheugen loos rondhangen.

59. Gegeven:

```

class mens {
public:
    char naam[30];
    mens* kind;
}; // mens

```

Deze “datastructuur” is bedoeld om een ouder-kind-kleinkind-...- relatie te representeren, een soort stamboom dus.

- Creëer een variabele van type `mens`, die een persoon geheten Wilhelmina voorstelt.
- Voeg haar dochter Juliana toe.
- Voeg haar kleindochter Beatrix toe.
- Voeg de vader van Wilhelmina, Willem, toe. Doe eventueel a opnieuw. Maak een tekening van de tot hier gemaakte structuur.
- Hoe kun je de naam van het kleinkind van Willem vinden?
- Stel dat je —in plaats van één— twee kinderen per mens wil toelaten. Hoe moet je de type-definitie dan aanpassen?
- Stel dat je de twee ouders van een kind ook zou willen weten. Hoe moet je de datastructuur dan aanpassen? Teken dit.
- Maakt het uit of in de definitie van de klasse `private` in plaats van `public` gebruikt wordt?

60. We hebben een lijst: $p \rightarrow 4 \rightarrow 9 \rightarrow 5$.

- Geef de noodzakelijke type-definities, declaraties, toekenningen etcetera om deze structuur te creëren.
- Schrijf een functie die in staat is een nieuw element achteraan de lijst toe te voegen (eerst voor deze speciale lijst, daarna algemeen).
- Hoe zou zo'n functie getest moeten worden?

61. (Tentamen 9 augustus 1996) Gegeven is het volgende type:

```

class vakje {
    int info;        // een getal
    vakje* rechts;  // wijst naar vakje er rechts naast
    vakje* onder;   // wijst naar vakje er onder
}; // vakje

```

Met behulp hiervan kan een lijst worden opgebouwd. Neem aan dat als het `info`-veld een waarde groter dan 0 bevat, de pointer `onder` NULL is, terwijl de pointer `rechts` NULL is als het `info`-veld een waarde kleiner dan 0 bevat. De andere pointer wijst dan naar het “volgende” vakje. Als de waarde 0 is, zijn beide pointers NULL. Een voorbeeld, met `ingang` van type `vakje*` (het vakje met -2 erin bevat hier een pointer (de “lijn” naar beneden) naar het vakje met 5 erin en een NULL-pointer):

```

ingang --> 3 --> -2
           |
           5 --> 10 --> 0

```

- Schrijf een C++-functie `voegtoe (ingang, getal)` die de `int` `getal` in een vakje vooraan de niet-lege lijst met `ingang` als `ingang` toevoegt. Denk eraan dat de werking afhangt van het al dan niet positief zijn van `getal`; neem aan dat `getal` niet 0 is.
- Schrijf een C++-functie `klapom (ingang)` die de waarde van het `info`-veld uit het eerste vakje van de niet-lege lijst `ingang` door zijn tegengestelde vervangt, en de pointers op de juiste wijze aanpast. In bovenstaand voorbeeld zou 3 vervangen worden door -3, en zou het vakje met -2 erin er onder komen te hangen.
- Schrijf een C++-functie `verwijder (ingang)` die het eerste vakje uit de lijst `ingang` verwijdert. Als de lijst leeg is hoeft er niets te worden gedaan.
- In de functies bij a, b en c staat in de heading steeds de parameter `ingang`. Deze heb je al dan niet “call by reference” gedeclareerd (met een `&`). Maakt het voor de werking van deze functies verschil uit of die `&` erbij staat? Leg ook uit wat er bij deze twee mogelijkheden precies gebeurt tijdens executie van de betreffende functie.
- Schrijf een integer C++-functie `som (ingang)` die de som oplevert van alle in de lijst `ingang` voorkomende `info`-velden. Als de lijst leeg is moet er 0 uitkomen.