

# Imperatief Programmeren

## TOETSPRACTICUM 1

**Opmerking:** Dit toetspracticum verschilt weinig van een normaal practicum. Je mag gewoon je literatuur en je aantekeningen raadplegen en ook de assistenten zijn aanwezig om je te helpen.

De enige verschillen zijn dat je alleen moet werken en dat je de opgaven pas aan het begin van de zitting hebt gekregen. Je hebt dus geen tijd gehad om de opdrachten voor te bereiden. Dat betekent dat we ook niet eisen dat je alle opgaven aan het eind van de zitting hebt afgerond. Je stuurt je werk op voor zover je opgaven hebt kunnen afronden. Zoals gebruikelijk wordt je oplossing onderworpen aan een aantal tests. Een door het controlesysteem geaccepteerde inzending levert in ieder geval één punt. Inspectie door een van de correctoren kan deze score verhogen tot maximaal drie punten per opgave.

Aan het eind van de zitting zal er gelegenheid zijn de opgave waar je mee bezig bent te beoordelen in te zenden, ook al is de uitwerking niet volledig. Ook zo'n incomplete uitwerking kan nog wat opleveren. Merk wel op dat we van iedereen hoogstens één onvolledige uitwerking accepteren. Deze opgave stuur je in onder de naam **NietAf**.

Het is niet nodig de opgaven te maken in de volgorde zoals ze hieronder zijn afgedrukt. Kies die opgaven waar je het meest vertrouwen in hebt.

Typ aan het begin van de zitting het commando `toets1`. Er wordt dan bij elke opgave een directory gemaakt; bij een aantal opgaven krijg je wat hulpfiles.

### **Opgave 1 Eerlijk delen!**

Opgavenaam: Delen

Moeder had een zakje dropjes gekocht. Ze had uitgerekend dat er bij gelijke verdeling onder haar drie kinderen twee dropjes voor haar zelf over zouden blijven. Toen ze thuis kwam bleken er twee vriendjes te zijn. De dropjes worden zo onder de vijf kinderen verdeeld dat ze allemaal evenveel krijgen. Ze hield er maar eentje voor haar zelf over.

We vragen ons af hoeveel dropjes er in het zakje hebben gezeten. Maak een programma dat een lijstje afdrukt van de mogelijke inhouden (ieder aantal op een nieuwe regel), waarbij we aannemen dat er niet meer dan 100 dropjes in het zakje hebben gezeten.

➤ lees verder ➤

## Opgave 2 Liefde op het Eerste Gezicht

Opgavenaam: Loeg

Ter gelegenheid van het lustrum van de televisiezender RTL4 wordt een speciale uitzending van het koppel-programma *'Liefde op het Eerste Gezicht'* gemaakt. Om het feest extra luister bij te zetten, worden van elke sexe 15 spelers uitgenodigd.

Voor de deelnemers aan deze toets die het genoemde TV-programma nog nooit hebben gezien, volgt hier de benodigde informatie. In eerste instantie zijn de kandidaten van de show keurig van elkaar gescheiden: een groep heren en een groep dames, beide met evenveel leden. Door een aantal vragen te beantwoorden, dan wel te omzeilen door middel van snedige opmerkingen, probeert elke kandidaat zich zo positief mogelijk te profileren.

Aan het eind van de show maakt elke heer via elektronische weg de dame van zijn voorkeur kenbaar; evenzo kiest elke dame haar favoriete heer. Als een dame en een heer elkaar hebben gekozen, we spreken dan van een *love-duo*; dan mogen ze op kosten van de organisatie een avondje uit.

Tot zover het TV-programma. We gaan nu naar het Java-programma. We stappen het verhaal binnen op het moment dat iedereen zijn keuze heeft gemaakt. Zowel de heren als de dames zijn genummerd van 0 tot en met 14. De keuzes zijn vastgelegd in een file met twee regels. Op de eerste regel staan achtereenvolgens de keuzes van de heren; op de tweede regel de keuzes van de dames.

De file `Loeg.java` bevat een aantal declaraties en een aanzet voor een methode om gegevens in te lezen. De inleesprocedure dient de gemaakte keuzes op te slaan in twee array's: `prefVanHeer` en `prefVanDame`. Bijvoorbeeld: `prefVanHeer[3]=11` betekent dat heer nummer 3 zijn oog heeft laten vallen op dame nummer 11.

Verder is een file met invoergegevens beschikbaar (`Loeg.inv`).

Schrijf een (efficiënt) programma dat een overzicht geeft van alle *love-duo's*. De volgorde dient naar opklimmend nummer van de betreffende heer te zijn.

Voorbeeld: bij de aangeleverde invoer (`Loeg.inv`) hoort de uitvoer

heer	dame
4	6
8	7
11	9
13	2

lees verder

## Opgave 3 De wet van Benford

Opgavenaam: Benford

(Deze opgave kun je zien als een vervolg op de opgave over grote gehele getallen uit een vorige practicumset)

Als je de blaadjes van een willekeurige boom (vol in blad) telt, hoe groot is dan de kans dat het eerste cijfer van dat aantal begint met een 1? Het blijkt dat die kans veel groter is dan, zoals je misschien zou verwachten, 1 op 9. Hetzelfde geldt voor de kans dat het aantal inwoners van een willekeurig gekozen plaats in Nederland met een 1 begint.

Benford ontdekte deze tegen-intuïtieve eigenschap bij het gebruik van logaritmentafels. Hij merkte op dat de pagina's met getallen die met een 1 beginnen aanmerkelijk meer bedruimd waren dan andere. Getalexperimenten leidden tot het volgende vermoeden (**De Wet van Benford**)

Laat  $c \in \{1, 2, \dots, 9\}$ .

De kans dat een willekeurig gekozen getal uit een oneindige verzameling getallen als eerste cijfer een  $c$  heeft is

$$10 \log\left(\frac{c+1}{c}\right)$$

Dit vermoeden kan natuurlijk niet waar zijn zonder eisen op te leggen aan de beschouwde verzameling (kijk maar eens naar de verzameling van alle gehele machten van 10). Voor een aantal typen verzamelingen is het wel te bewijzen, maar dat zou hier te ver voeren.

Gevraagd wordt nu dit vermoeden te toetsen met een eindige deelverzameling van de verzameling van gehele machten van 2. In onderstaande stappen waarin het definitieve programma wordt ontwikkeld zijn de onderdelen 1 en 3 alleen bedoeld om je op weg te helpen. Het gaat uiteindelijk om het afdrukken van een tabel zoals in onderdeel 4 wordt beschreven. Dat is ook de versie die je ter beoordeling in moet sturen.

1. Vraag de gebruiker om de hoogste macht van 2 die meegenomen moet worden en schrijf in je hoofdprogramma een programma-fragment dat een lijst met de eerste machten van 2 afdrukt ( $2^0, 2^1, \dots$ ) tot en met de door de gebruiker aangegeven macht. Je mag aannemen dat de gebruiker een niet-negatief geheel getal invoert.
2. Voeg aan je klasse `BigInt` een methode toe die het eerste cijfer van het gerepresenteerde getal oplevert.
3. Vul je hoofdprogramma zo aan, dat naast elke tweemacht ook zijn eerste cijfer wordt afgedrukt. Test je programma.
4. Druk een tabel af waarbij voor iedere  $c \in \{1, \dots, 9\}$  het aantal voorkomens van  $c$  als eerste cijfer wordt gegeven. Druk eveneens af welke frequentie volgens de wet van Benford verwacht mag worden (verwachte frequentie = kans  $\times$  totaal aantal). Kies hiervoor de geheeltallige afronding van de verwachte waarde. Zie het voorbeeld voor de lay out.

lees verder

Voorbeeld:

De wet van Benford

Geef de hoogste macht: <99>

cijfer	freq	benford
1	30	30
2	17	18
3	13	12
4	10	10
5	7	8
6	7	7
7	6	6
8	5	5
9	5	5

## Opgave 4 Schijven in het platte vlak

Opgavenaam: Schijven

Een aantal cirkelschijven wordt op een tafel geworpen. De vraag is nu hoeveel van die schijven helemaal vrij liggen, dat wil zeggen dat ze op geen enkel punt overlappen met enige andere schijf.

Een schijf wordt gerepresenteerd door zijn middelpunt en zijn straal. We nemen in deze opgave aan dat de coördinaten van het middelpunt geheeltallig zijn en dat de straal een natuurlijk getal is. We beschouwen de schijf als een gesloten cirkel, dat wil zeggen dat de rand van de schijf bij de schijf wordt gerekend.

Aan het begin van dit practicum is de file `Schijven.inv` voor je klaar gezet. Op de eerste regel van deze file staat een natuurlijk getal  $n$ , het aantal schijven. Daarna volgen er  $n$  regels met steeds drie gehele getallen, van elkaar gescheiden door een spatie. Dit zijn resp. de  $x$ -coördinaat, de  $y$ -coördinaat en de lengte van de straal van de betreffende cirkel.

De bedoeling is om een programma te maken dat bij een invoerfile (van het formaat zoals hierboven beschreven) afdruckt hoeveel vrijliggende schijven er zijn. Onder een vrijliggende schijf verstaan we een schijf die geen enkel punt gemeen heeft met enige andere schijf. Zie het voorbeeld hieronder voor de layout van de uitvoer.

We vragen van je om het programma zo te schrijven, dat de invoerfile als parameter bij de aanroep moet worden meegegeven. Verder verwachten we dat je een klasse introduceert die een cirkelschijf representeert en die (voor deze opgave) nuttige functionaliteit bevat.

Voorbeeld: bij de gegeven invoerfile hoort de uitvoer

Schijven

Aantal vrijliggende schijven: 3

➤ einde