Extra opgaven - Python

Thibaut Deliever

13 oktober 2020

Inhoudsopgave

Inleiding

In dit document vindt u extra opgaven terug die te vinden waren op het externe platform "Dodona". Elk hoofdstuk gaat over een ander element dat wordt ingeoefend. Binnen elk hoofdstuk zal worden weergegeven wat de moeilijkheidsgraad is per oefening. Op de github repository kan u mijn of Dieter's oplossing terugvinden van deze opgaven. Deze repository is te bereiken via deze link `http://github.com/kulak-informatica`.

1 Expressies

Oefeningen	Moeilijksheidsgraad
Hello World	☆☆☆
Boekenwinkel	★☆☆
Protonen	★☆☆
IBAN	★☆☆
Leren Delen	★★☆
Patroontje	★★★

1.1 Hello World

Een Hello world-programma is een eenvoudig computerprogramma dat niets anders doet dan de tekst "Hello worldöp het scherm tonen.

Een dergelijk programma wordt meestal als eerste voorbeeld gebruikt in een cursus programmeren. Het wordt tevens gebruikt om na te gaan of de programmeeromgeving functioneert.

1.2 Boekenwinkel

Een boek kost in de winkel € 24.95, maar boekenwinkels krijgen 40% korting bij inkoop. Voor het verschepen van boeken gelden volgende tarieven:

- € 3 voor het eerste boek;
- € 0.75 voor ieder volgende boek.

Bereken hoeveel de winkel betaalt voor 60 boeken.

1.3 Protonen

Terwijl een Duitse supercomputer onlangs een simulatie heeft uitgevoerd en heeft geschat dat er ongeveer 500 miljard melkwegstelsels binnen de observeerbare ruimte liggen, wordt dit aantal volgens een conservatievere schatting op ongeveer 300 miljard geschat.

Aangezien het aantal sterren in een melkwegstelsel kan oplopen tot 400 miljard, schat men het aantal sterren op $1.2 \cdot 10^{23}$. Gemiddeld weegt elke ster ongeveer 10^{35} gram. De totale massa zou dus ongeveer 10^{58} gram bedragen. Aangezien elke gram materie ongeveer 10^{24} protonen bevat (of ongeveer hetzelfde aantal waterstofatomen aangezien een waterstofatoom slechts 1 proton heeft), zou het totale aantal waterstofatomen ongeveer 10^{86} bedragen.

Opgave

Print het totaal aantal protonen in de observeerbare ruimte als een getal zonder exponent.

Uitvoer

```
Protonen:  100000000000000000000000000000000000000000000000
000000000000000000000000000000000000000
```

1.4 IBAN

Wanneer je tijdens het online bankieren geld wil overmaken, dan moet je het IBAN-rekeningnummer van de begunstige opgeven. Opdat je geen fouten zou typen in het rekeningnummer, wordt het IBAN-nummer gevalideerd. Je kun dus enkel geld overschrijven naar een geldig rekeningnummer.

De wijze waarop de validatie gebeurt, is wereldwijd hetzelfde:

1. Controleer of het getal op 3e en 4e positie tussen 2 en 98 ligt.
2. Valideer de samenstelling. Voor België is dit cckk BBBC CCCC CCKK met:
   - cc = landcode,
   - kk = het controlegetal van het volledige IBAN-nummer,
   - B = bankcode,
   - C = rekeningnummer,
   - K = controlegetal (deel van het nationale rekeningnummer).
3. Verplaats de eerste 4 karakters naar het einde.
4. Vervang elke letter door 2 cijfers, waarbij A = 10, B = 11, ... , Z = 35.
5. Bereken dan het getal modulo 97. Het resultaat moet gelijk zijn aan x.

Opgave

Het resultaat x van de bewerking uit de laatste stap is wereldwijd hetzelfde. Als je weet dat BE68 5390 0754 7034 een geldig IBAN-nummer is, bereken dan het resultaat x dat je in de laatste validatiestap moet uitkomen.

1.5 Leren delen

Het aanleren van de deling in de lagere school gebeurt in verschillende fasen.

- Eerst leert men opgaande delingen zoals 8 ÷ 2.

- Vervolgens leert men over delingen met een gehele oplossing en rest zoals $7 \div 2 = 3$ met rest 1.
- Ten slotte komen staartdelingen aan bod.

Opgave

In deze oefening leggen we de focus op de tweede fase. Je deelt 21 door 6 en toont samen met de oplossing ook de rest. Wil je iets bijleren, gebruik dan in je code

- geen 3;
- geen typecasting naar een ander gegevenstype dan string.

Uitvoer

```
21/6 = 3 met rest 3
```

1.6 Patroontje

Laat volgende uitvoer genereren: `12.012.012.012.012.0`

Wil je iets bijleren, gebruik dan in je code geen '.' en slechts 1 typecasting.

2 Variabelen

Oefeningen	Moeilijksheidsgraad
Oppervlakte cirkel	★☆☆
Wet van Coulomb	★★☆
Avogadro	★★☆
Transformaties	★★☆
Wisselgeld	★★★

2.1 Oppervlakte cirkel

Schrijf een programma dat de straal $r \in \mathbb{R}_0^+$ van een cirkel vraagt. Het programma toont de oppervlakte van een cirkel met straal r.

Voor het geval je het vergeten bent, de formule voor de oppervlakte van een cirkel met straal r is $O = \pi \cdot r^2$ met $\pi = 3.14159$.

Invoer

```
2
```

Uitvoer

```
De oppervlakte van een cirkel met straal 2.0 is 12.56636
```

Invoer

```
4.2222
```

Uitvoer

```
De oppervlakte van een cirkel met straal 4.2222 is
56.0050396044156
```

2.2 Wet van Coulomb

De **wet van Coulomb**, genoemd naar de Franse natuurkundige Charles-Augustin de Coulomb, beschrijft de kracht die twee elektrische puntladingen op elkaar uitoefenen. Deze kracht is recht evenredig met elk van de ladingen en omgekeerd evenredig met het kwadraat van de onderlinge afstand van de ladingen. Als de ladingen beide positief of beide negatief zijn, oefenen zij een afstotende kracht op elkaar uit, en wordt de kracht als positief gerekend. Zijn de ladingen tegengesteld van teken, dan is de kracht een aantrekking, en wordt zij negatief gerekend.

In deze oefening gaan we uit van twee positieve puntladingen en Q_1 en Q_2. De wet van Coulomb wordt dan gegeven door:

$$F_C = k \cdot \frac{Q_1 \cdot Q_2}{r^2}$$

In het vacuüm is de constante k gelijk aan k_0:

$$k_0 = 8.99 \cdot 10^9 \frac{N \cdot m^2}{C^2}$$

Opgave

Gegeven zijn twee puntladingen $Q_1 = 2.0\mu C$ en $Q_2 = 1.0\mu C$. De twee puntladingen worden in het vacuüm opgesteld op een afstand r , uitgedrukt in cm, die je aan de gebruiker vraagt. Het programma schrijft de coulombkracht F_C uit.

Invoer

```
3.810210093889077
```

Uitvoer

```
12.384881084179142
```

Invoer

```
7.996864888080065
```

Uitvoer

```
2.8115782213051665
```

2.3 Avogadro

De **constante van Avogadro**, N_A , ook het getal van Avogadro genoemd, is een fysische constante die de verhouding tussen het aantal deeltjes en de hoeveelheid stof aangeeft. De constante is gedefinieerd als het aantal deeltjes per mol^{-1}, met als eenheid mol.

$$N_A = 6.020 \cdot 10^{23} \cdot \text{mol}^{-1}$$

Als je weet dat $m(X)$ de massa van een stof is en $n(X)$ het aantal mol deeltjes van een stof is, dan wordt de hoeveelheid stofdeeltjes $N(X)$ gegeven door:

$$N(X) = n(X) \cdot N_A$$

De molaire massa $M(X)$ van een stof (X) wordt dan gegeven door:

$$M(X) = \frac{m(X)}{n(X)}$$

Opgave

In deze oefening werken we met zwavel (S). Voor zwavel geldt:

$$M(S) = 32.06 \frac{\text{g}}{\text{mol}}$$

Je programma vraagt aan de gebruiker een aantal deeltjes S, uitgedrukt in mol. Laat je programma achtereenvolgens de massa en het aantal deeltjes horende bij de massa van S uitschrijven.

Invoer

```
9.206826283274985
```

Uitvoer

```
295.17085064179605
5.54250942253154e+24
```

Invoer

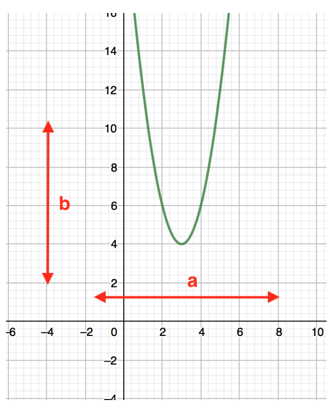
```
0.9206030924140651
```

Uitvoer

```
29.51453514279493
5.542030616332671e+23
```

2.4 Transformaties

Gegeven is een tweedegraadsfunctie $f(x) = 2(x-3)^2 + 4$. De grafiek van f ziet er als volgt uit:



Figuur 1: grafiek van f

De grafiek van de functie f kan je zowel horizontaal als verticaal verschuiven. Dat doe je als volgt:

- $f(x-a)$ met $a \in \mathbb{R}$ verschuift de grafiek van f in horizontale richting met a.
- $f(x)+b$ met $b \in \mathbb{R}$ verschuift de grafiek van f in verticale richting met b.

Opgave

Je programma vraagt achtereenvolgens a en b op aan de gebruiker. Je toont onder elkaar:

- Het functievoorschrift van f.
- Het functievoorschrift van $g(x) = f(x-a)+b$.

Invoer

```
24
31
```

Uitvoer

$f(x) = 2(x-3)^2 + 4$
$g(x) = 2(x-27)^2 + 35$

Invoer

```
-18
-40
```

Uitvoer

$f(x) = 2(x-3)^2 + 4$
$g(x) = 2(x--15)^2 + -36$

2.5 Wisselgeld

Schrijf code die een hoeveelheid eurocenten classificeert als een combinatie van grotere geldstukken. Je programma gebruikt stukken van € 1, 50 cent, 20 cent, 10 cent, 5 cent, 2 cent en 1 cent.

Opgave

Je vraagt een bepaald bedrag in eurocenten aan de gebruiker. Vervolgens bepaal je hoeveel stukken van € 1 er in het gegeven bedrag passen. Dan hoeveel stukken van 50 cent er in het restbedrag zitten nadat de stukken van € 1 eruit genomen zijn, dan de stukken van 50 cent, dan de stukken van 20 cent, enz. Het programma toont het bedrag uitgedrukt in het minimaal aantal muntstukken waarin je het bedrag kan wisselen.

Invoer

```
55
```

Uitvoer

```
55 cent kan je omwisselen in 2 muntstukken
```

Invoer

```
199
```

Uitvoer

```
199 cent kan je omwisselen in 7 muntstukken
```

3 Variabelen +

Oefeningen	Moeilijksheidsgraad
Tijdmeting op Mars	★★★
De krekel als thermometer	★★★
De gestopte klok	★★★
Graansilo vullen	★★★
LIDAR	★★★

3.1 Tijdmeting op Mars

Op 6 augustus 2012 landde de rover Curiosity in de Gale krater op Mars als deel van de Mars Science Laboratory (MSL) missie van NASA. Op vandaag (11 oktober 2020) duurt de missie van Curiosity op het Marsoppervlak reeds 2908 sol (2988 dagen). De term sol wordt door astronomen gebruikt om de duur van een zonnedag op Mars aan te duiden. Een gemiddelde Marsdag — of sol — duurt 24 uur, 39 minuten en 35,244 seconden. Het ogenblik waarop een landingsmodule het oppervlak van Mars bereikt, wordt gebruikt als referentiepunt om de duur van de missie te bepalen. Dit referentiepunt wordt "sol 0"genoemd, waardoor eenvoudigweg het aantal verstreken Marsdagen (sol) sinds de landing kan geteld worden.

De invoer bestaat uit een natuurlijk getal dat een aantal Marsdagen (sol) uitdrukt. De uitvoer drukt het gegeven aantal sol uit in Aardse dagen, uren, minuten en seconden. Bij deze omzetting moet gebruikgemaakt worden van het feit dat 1 sol gelijk is aan 24 uur, 39 minuten en 35,244 seconden. Het formaat waarin de uitvoer moet uitgeschreven worden, kan afgeleid worden uit onderstaand voorbeeld. Hierbij moeten alle waarden als natuurlijke getallen uitgeschreven worden, en moet afkapping gebruikt worden om het aantal seconden uit te drukken als een natuurlijk getal.

Invoer

```
2908
```

Uitvoer

```
2908 sol = 2987 dagen, 22 uren, 40 minuten en 9 seconden
```

3.2 De krekel als thermometer

Krekels tjirpen door hun vleugels langs elkaar te strijken. Bij de meeste soorten zijn het enkel de mannetjes die het zo bekende geluid voortbrengen om zo partners te kunnen aantrekken. Wanneer je rustig aan het luisteren bent naar het rustgevende geluid van tjirpende krekels, zou je dus eigenlijk de bedenking moeten maken dat je aan het luistervinken bent naar een paringsritueel waarmee degene die de serenade brengt als enige doel heeft om wellustige mannetjeskrekels af te schrikken en geïnteresseerde vrouwtjes het hof te maken.

De idee dat het tellen van de tjirpgeluiden die krekels voortbrengen ook kan dienen als een informele manier om de temperatuur te bepalen is echter niet nieuw. Het werd oorspronkelijk beschreven in 1897 door de natuurkundige Amos Dolbear, in een artikel met als titel "De krekel als thermometer". Daarin stelde Dolbear aanvankelijk dat de buitentemperatuur een belangrijke bepalende factor is voor de frequentie waarmee krekels tjirpen. Doorheen de jaren werd zijn manier om naar dit verband te kijken echter omgekeerd — mensen tellen nu het aantal tjirpgeluiden om daarmee de temperatuur te bepalen, eerder dan naar een thermometer te kijken om te voorspellen hoeveel tjirpgeluiden ze zullen horen. De wet van Dolbear geeft het verband aan onder de vorm van volgende formule, die aangeeft hoe de temperatuur T_F in graden Fahrenheit kan geschat worden op basis van het aantal gehoorde tjirps per minuut N_{60}:

$$T_F = 50 + \left(\frac{N_{60} - 40}{4}\right)$$

Deze formule kan ook herschreven worden om de temperatuur in graden Celcius (°C) te bepalen:

$$T_C = 10 + \left(\frac{N_{60} - 40}{7}\right)$$

Bovenstaande formules worden uitgedrukt in termen van gehele getallen, zodat ze makkelijker kunnen onthouden worden — ze zijn niet direct geschikt om een exacte temperatuursbepaling te doen.

De invoer bestaat uit het aantal waargenomen tjirps per minuut $N_{60} \in \mathbb{N}$. De uitvoer is een regel die de tekst temperatuur (Fahrenheit): T_F bevat, waarbij de temperatuur T_F in graden Fahrenheit aangeeft. Hierbij wordt T_F bepaald volgens de wet van Dolbear voor een gegeven aantal tjirps per minuut N_{60} zoals die wordt ingelezen uit de invoer. Een tweede regel die dezelfde temperatuur T_C weergeeft, maar dan uitgedrukt in graden Celsius.

Invoer

```
43
```

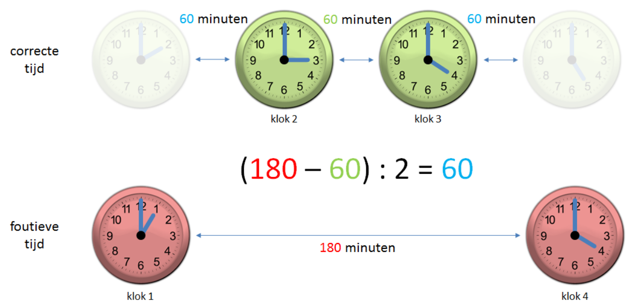
Uitvoer

```
temperatuur (Fahrenheit):  50.75
temperatuur (Celsius):  10.428571428571429
```

3.3 De gestopte klok

Het enige uurwerk dat Andrea in huis heeft, is een grote staande klok. Op een dag is ze vergeten om de klok op te winden, waardoor de klok gestopt is met tikken. Enkele dagen later reist ze naar de stad om er te gaan lunchen bij een vriendin wiens klok altijd de correcte tijd aangeeft. Onmiddellijk nadat ze thuiskomt, maakt ze een eenvoudige berekening waarmee ze haar klok terug juist kan zetten.

Hoe is ze daarin geslaagd zonder de exacte reistijd te kennen tussen haar huis en dat van haar vriendin?



Figuur 2: De gestopte klok

De oplossing ligt eigenlijk vrij voor de hand. Vooraleer haar huis te verlaten, windt Andrea haar staande klok terug op en onthoudt ze de tijd die de klok bij vertrek aangeeft. Daarna onthoudt ze ook de correcte tijd die de klok in het huis van haar vriendin aangeeft, zowel bij aankomst als bij vertrek. Als ze terug thuiskomt, kan ze op basis van de tijd die haar klok aangeeft, bepalen hoeveel tijd de volledige trip geduurd heeft. Daar trekt ze de tijd van af die ze in het huis van haar vriendin heeft doorgebracht, en deelt het resultaat door twee om de reistijd tussen haar huis en dat van haar vriendin te weten te komen. Als ze deze tijdsduur optelt bij de tijd die de klok aangaf als ze bij het huis van haar vriendin vertrok, dan bekomt ze de huidige tijd waarmee ze haar eigen klok kan instellen.

De invoer bestaat uit 8 natuurlijke getallen, die elk op een afzonderlijke regel staan. Elk paar opeenvolgende getallen h ($0 \leq h \leq 24$) en m ($0 \leq m \leq 60$) correspondeert met de tijdsaanduiding h:m op een 24-uursklok. De vier tijdsaanduidingen geven achtereenvolgens de tijd aan op het ogenblik dat Andrea

- vertrekt vanaf haar eigen huis
- aankomt bij het huis van haar vriendin
- terug vertrekt vanaf het huis van haar vriendin
- terug aankomt bij haar eigen huis

De uitvoer bestaat uit twee regels die de natuurlijke getallen h ($0 \leq h \leq 24$) en m ($0 \leq m \leq 60$) bevatten die corresponderen met de correcte tijd h:m op een 24-uursklok wanneer Andrea terug aankomt bij haar eigen huis. Je mag ervan uitgaan dat Andrea niet langer dan 24 uur van huis is weggeweest.

Invoer

```
13
0
15
0
16
0
16
0
```

Uitvoer

```
17
0
```

3.4 Graansilo vullen

Een landbouwer heeft een veld van b meter breed en l meter lang. Dit veld levert c kubieke meter graan per hectare op (1 hectare = 10.000 vierkante meter). De landbouwer heeft een aantal cilindervormige graansilo's waarin hij zijn oogst opslaat. Elke graansilo heeft een straal van r meter en hoogte van h meter.

De invoer bestaat uit de waarden van b, l, c, r en $h \in \mathbb{R}$, in die volgorde en elk op een afzonderlijke regel. De uitvoer bestaat uit twee regels. Op de eerste regel staat het aantal graansilo's dat de landbouwer nodig heeft om zijn oogst op te slaan. Op de tweede regel staat de hoogte van de oogst in de laatste graansilo, die eventueel slechts gedeeltelijk gevuld is.

Invoer

```
503.3
623.4
5.5
2.1
5.6
```

Uitvoer

```
3
1.255694726337511
```

Invoer

```
3.1415926535897931
16
100000
4
10
```

Uitvoer

```
1
10.0
```

3.5 LIDAR

Lidar (LIght Detection And Ranging) is een technologie die de afstand tot een object of een oppervlak bepaalt door middel van laserpulsen. De techniek is vergelijkbaar met radar, waarbij radiogolven gebruikt worden in plaats van licht. De afstand wordt bepaald door het meten van de tijd die verstrijkt tussen het uitzenden van een puls en het opvangen van de reflectie ervan.

Lidar werkt volgens hetzelfde principe als radar: een signaal wordt uitgezonden en de reflectie ervan wordt enige tijd later weer opgevangen. Het verschil tussen lidar en radar is dat lidar gebruik maakt van een laserstraal terwijl radar gebruik maakt van radiogolven. Hierdoor kunnen met lidar veel kleinere objecten gedetecteerd worden dan met radar. De golflengte van radiogolven ligt immers rond de 1 cm, terwijl deze van laserlicht tussen de 10 μm en 250 nm ligt. Bij deze golflengte zullen kleine objecten de golven beter reflecteren. De formule die gebruikt wordt voor het bepalen van de afstand tot een object is:

$$d = \frac{c \cdot t}{2 \cdot n}$$

Hierbij stelt d de afstand (in meter) tot het object voor, t de tijd (in seconden) die de laserstraal nodig heeft om heen en terug te reizen, c de lichtsnelheid in vacuüm (299792458 m/s) en n de refractieindex. In lucht is de refractieindex onder standaardomstandigheden gelijk aan 1,000277.

De invoer is een getal $t \in \mathbb{N}$ dat de tijd in **nanoseconden** voorstelt. 1 nanoseconde is gelijk aan 10^{-9} seconden. De uitvoer is de d afstand (in meter) tot het object (waarbij we standaardomstandigheden van de lucht veronderstellen).

Invoer

```
100
```

Uitvoer

```
14.985471924276975 meter
```

4 Eenvoudige Functies

Oefeningen	Moeilijksheidsgraad
Absolute waarde	★☆☆
Sommetjes	★★☆
Pythagoras	★★★

4.1 Absolute waarde

De **absolute waarde** van een getal $x \in \mathbb{R}$ wordt aangegeven door $|x|$ en definieert men als volgt:

$$|x| = \begin{cases} x & \text{als } x \geq 0 \\ -x & \text{als } x < 0 \end{cases}$$

Een eigenschap van de absolute waarde gaat als volgt:

$$\forall x, y \in \mathbb{R} : ||x| - |y|| \leq |x - y|$$

Geef een empirisch bewijs door deze eigenschap te testen voor twee willekeurige getallen x , $y \in \mathbb{R}$.

Opgave

Jouw programma vraagt twee willekeurige getallen als invoer. Vervolgens bereken je zowel het linker- als rechtlid van de ongelijkheid. Je toont het resultaat op een nette geformatteerde manier. Toon maximaal 4 cijfers na de komma.

Om speciale tekens te tonen, zoek je bijvoorbeeld op unicode-search.net naar alle karakters die iets te maken hebben met less. Hieronder een voorbeeld van het gebruik van een unicode-naam.

```
>>> print('\N{LESS-THAN OR EQUAL TO}')
≤
```

Invoer

```
42.06826283274985
13.510020196930185
```

Uitvoer

```
28.5582 ≤ 28.5582
```

Invoer

```
37.18977709291377
-3.0396936602994487
```

Uitvoer

```
34.1501 ≤ 40.2295
```

4.2 Sommetjes

De invoer van het programma zijn twee natuurlijke getallen $a, b \in \,]0, 20]$. De uitvoer van het programma zijn 5 sommetjes van de vorm:

$$10^i \cdot a + 10^i \cdot b = c \quad \text{met} \quad i = 0, \ldots, 4$$

Zorg voor een geformatteerde uitvoer.

Invoer

```
18
20
```

Uitvoer

```
     18   +   20       =   38
    180   +   200      =   380
   1800   +   2000     =   3800
  18000   +   20000    =   38000
 180000   +   200000   =   380000
```

Invoer

```
9
8
```

Uitvoer

```
     9   +   8       =   17
    90   +   80      =   170
   900   +   800     =   1700
  9000   +   8000    =   17000
 90000   +   80000   =   170000
```

4.3 Pythagoras

De **stelling van Pythagoras** is een wiskundige stelling die haar naam dankt aan de Griekse wiskundige Pythagoras. 'Zijn' stelling was overigens alleen maar nieuw voor de Grieken. In Soemer was het resultaat al veel langer bekend en ook in Babylonië en het oude Egypte werd ze al eerder toegepast (met name de verhouding $a = 3; b = 4; c = 5$ werd al vroeg gebruikt om rechte hoeken uit te meten, zoals dat tot op de dag van vandaag door sommigen nog wordt gedaan). Naast kennis van de stelling om haar toe te kunnen passen, is ook het leveren van een bewijs belangrijk. Wat dat betreft waren de Grieken (Pythagoras of een van zijn leerlingen) wel de eersten. Zij wisten niet alleen dat de stelling waar was, maar konden ook in algemene termen (abstracties) aantonen waarom zij waar was.

Stelling

Stelling In een rechthoekige driehoek is de som van de kwadraten van de lengtes van de rechthoekszijden gelijk aan het kwadraat van de lengte van de schuine zijde.

Noemt men de lengten van rechthoekszijden (de zijden die aan de hoek van 90° liggen) a en b, en de lengte van de schuine zijde (de zijde die niet aan de rechte hoek grenst, ook wel "hypotenusa"genoemd) c, dan wordt de bekende wiskundige vorm van de stelling als volgt geschreven:

$$a^2 + b^2 = c^2$$

Opgave

De invoer van het programma bestaat uit twee getallen $a, b \in \mathbb{R}_0^+$. Het programma berekent de lengte van de schuine zijde . Je toont het resultaat op een nette geformatteerde manier. Rond steeds af tot twee getallen na de komma.

Invoer

```
92.06826
16.1
```

Uitvoer

```
In een rechthoekige driehoek met rechthoekszijden a = 92.07
en b = 16.10 is de schuine zijde 93.47
```

Invoer

```
3.0
78.0
```

Uitvoer

```
In een rechthoekige driehoek met rechthoekszijden a = 3.00
en b = 78.00 is de schuine zijde 78.06
```

5 Eenvoudige Functies +

Oefeningen	Moeilijksheidsgraad
Gevoelstemperatuur	★★★
De diatomist	★★★
Vis viva	★★★

5.1 Gevoelstemperatuur

Je kent zeker het fenomeen dat je bij felle wind de koude des te meer aanvoelt dan wanneer het windstil is. Dit fenomeen noemt men met een Engelse term wind chill. Het zal je ook niet verbazen dat er verschillende methoden bestaan om te bepalen wat nu juist de gevoelstemperatuur is voor een gegeven luchttemperatuur en windsnelheid. Een van die methoden is de JAG/TI-methode (Joint Action Group on Weather Indices). Deze in Canada ontwikkelde methode is gebaseerd op het warmtetransport van het lichaam naar de huid. Als de temperatuur in °C op 1,50 meter hoogte gegeven wordt door T en de gemiddelde windsnelheid in de afgelopen tien minuten in m/s op 10 meter hoogte gegeven wordt door W , dan kan je de gevoelstemperatuur berekenen met de volgende formule:

$$13,12 + 0,6215\,T + (0,3965\,T - 11,37)(3,6\,W)^{0,16}$$

De gevoelstemperatuur geldt voor een gezond, volwassen en wandelend persoon van gemiddelde lengte. De zon speelt geen rol in de berekeningsmethode, maar bij zonnig weer voelt het minder koud aan dan de berekende gevoelstemperatuur doet vermoeden. Ook wanneer je met de wind in de rug wandelt, zal het minder koud aanvoelen.

De invoer bestaat uit de volgende twee reële getallen, die elk op een afzonderlijke regel staan:

- de luchttemperatuur T in °C
- de windsnelheid W in km/u

De uitvoer bestaat uit één reëel getal dat de gevoelstemperatuur geeft volgens bovenstaande formule. Je hoeft je resultaat niet af te ronden.

Invoer

```
10.0
5.0
```

Uitvoer

```
9.755115709161835
```

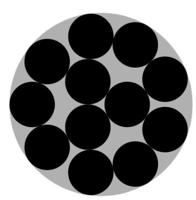
5.2 De diatomist

Klaus Kemp is de enige moderne beoefenaar van een verloren gewaande Victoriaanse kunstvorm — het schikken van diatomeeën in minuscule, oogverblindende patronen die wat weg hebben van microscopische mozaïeken of glas-in-loodramen. Diatomeeën zijn eencellige wieren met een extern skelet van kiezel (siliciumdioxide, SiO2).

De klasse van de diatomeeën telt ongeveer 100.000 verschillende soorten, waarvan de meeste variëren in grootte van 10 tot 100 micrometer. Op het einde van de 19e eeuw ontstond het ambacht om ze door professionele microscopisten in allerlei patronen te laten schikken voor vermogende klanten. Hoe ze er precies in slaagden om dat te doen is onbekend — ze namen hun geheimen immers mee in hun graf. Kemp deed ongeveer acht jaar over het perfectioneren van zijn eigen techniek, die erin bestaat om de vormen over een periode van enkele dagen op een heel nauwkeurige manier te schikken in een laagje lijm.

Opgave

In deze opgave vragen we ons af hoeveel kleinere cirkels met straal $r \in \mathbb{R}^+$ kunnen geschikt worden in een grotere cirkel met straal $R \in \mathbb{R}^+$, zonder dat de kleinere cirkels elkaar overlappen. Zo passen er bijvoorbeeld maximaal 12 kleinere cirkels in de grotere cirkel uit onderstaande figuur.



Figuur 3: Illustratie van het probleem

Er werd vooralsnog nog geen algemeen antwoord op deze vraag gevonden, maar het maximaal aantal kleinere cirkels n kan op de volgende manier benaderd worden:

$$n = \left\lfloor 0.83\frac{R^2}{r^2} - 1.9 \right\rfloor$$

Hierbij is $\lfloor x \rfloor$ het grootste geheel getal dat niet groter is dan $x \in \mathbb{R}$. De invoer bevat twee getallen r en $R \in \mathbb{R}^+$ — elk op een afzonderlijke regel — die respectievelijk de straal van een kleinere en een grotere cirkel aangeven. Er geldt met andere woorden dat $r \leq R$ (dit hoef je zelf niet expliciet te testen in je eigen code). De uitvoer is een regel die aangeeft hoeveel kleinere cirkels er maximaal in de grotere cirkel kunnen geschikt worden, en de bedekkingsgraad van de grotere cirkel die daarmee bekomen wordt. Voor het schatten van het maximaal aantal kleinere cirkels moet bovenstaande formule gebruikt worden.

De bedekkingsgraad moet uitgeschreven worden als een reeël getal dat afgerond is tot op twee cijfers na de komma.

Invoer

```
2.38
10.14
```

Uitvoer

```
13 kleine cirkels bedekken 71.62% van de grote cirkel
```

5.3 Vis viva

Vis viva (Latijn voor levende kracht) is een achterhaalde wetenschappelijke theorie die kan beschouwd worden als voorloper van de wet van behoud van energie. Ze gaf voor het eerst een beschrijving van de kinetische energie, waarbij de levende kracht verwijst naar alle kinetische energie in een geïsoleerd systeem.

Vandaag de dag is vis viva opgenomen en vervangen door de moderne theorie van energie. In de sterrenkunde leeft de naam echter nog voort in de **vis-vivavergelijking**: volgens de klassieke (Newtoniaanse) hemelmechanica draaien satellieten in een ellipsvormige baan rond de Aarde.

De vis-vivavergelijking legt een verband tussen de grote as a van de ellipsvormige baan, de afstand r van de satelliet tot het middelpunt van de Aarde en de snelheid v van de satelliet ten opzichte van de Aarde:

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right)$$

De constante μ is de geocentrische gravitatieconstante. Een benadering van deze constante kan berekend worden als het product van de gravitatieconstante G en de massa M van de Aarde uitgedrukt in kilogram:

$$\mu = G \cdot M$$

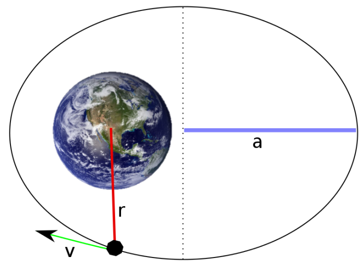
Er bestaan echter alternatieve methoden om de waarde van nauwkeuriger te meten. In deze opgave gebruiken we de volgende nauwkeurige meetwaarde:

$$\mu = 398600{,}4418 \cdot 10^9\, m^3 s^{-2}$$

Als de lengte van de grote as a van een elliptische baan gekend is dan kan daarmee de **periode** p van de satelliet bepaald worden (uitgedrukt in seconden):

$$p = 2\pi\sqrt{\frac{a^3}{\mu}}$$

De periode p is de tijd die de satelliet nodig heeft om één omwenteling rond de Aarde te maken.



Figuur 4: Illustratie van het probleem

De invoer bevat twee reële getallen die elk op een afzonderlijke regel staan:

- de afstand r tussen een satelliet en het middelpunt van de Aarde (uitgedrukt in meter)
- de snelheid v van de satelliet ten opzichte van de Aarde (uitgedrukt in meter/seconde)

Met deze gegevens kan de lengte a (in meter) berekend worden van de grote as van de elliptische baan waarin de satelliet rond de Aarde draait. Hiervoor kan de vis-vivavergelijking herschreven worden als:

$$a = \frac{\mu r}{2\mu - rv^2}$$

De periode p van de satelliet (uitgedrukt in seconden) wordt dan bekomen als:

$$2\pi\sqrt{\frac{a^3}{\mu}}$$

Er moeten drie regels uitgeschreven worden:

- de lengte van de grote as a, uitgedrukt in meter
- de lengte van de periode, uitgedrukt in seconden
- de lengte van de periode, uitgedrukt in een geheel aantal dagen d, uren u en minuten m die volledig binnen de periode passen; hierbij moet gelden dat $0 \leq u < 24$ en dat $0 \leq m < 60$

Bekijk onderstaande voorbeelden om te achterhalen hoe de uitvoer precies moet uitgeschreven worden.

Invoer

```
6792000
7658
```

Uitvoer

```
grote as:  6787166.808499204 meter
periode:  5564.7257424392155 seconden
periode:  0 dagen, 1 uren en 32 minuten
```

Invoer

```
35785400
3580.9
```

Uitvoer

```
grote as:  42160215.133579694 meter
periode:  86151.96905571753 seconden
periode:  0 dagen, 23 uren en 55 minuten
```

Invoer

```
7104000
7485
```

Uitvoer

```
grote as:  7093371.63898765 meter
periode:  5945.52283951033 seconden
periode:  0 dagen, 1 uren en 39 minuten
```

Invoer

```
400000000
977.75
```

Uitvoer

```
grote as:  384375790.60599077 meter
periode:  2371619.541180138 seconden
periode:  27 dagen, 10 uren en 46 minuten
```

6 Condities

Oefeningen	Moeilijksheidsgraad
Crocodile Paradox	★☆☆
Trolleyprobleem	★☆☆
Schaal van Saffir-Simpson	★★☆
Blad steen schaar	★★☆
Risk	★★★

6.1 Crocodile Paradox

Een krokodil, die een kind heeft gestolen, belooft de vader dat zijn kind zal worden teruggegeven als en alleen als hij correct voorspelt wat de krokodil zijn volgende actie zal zijn. De krokodil denkt dat de vader zal zeggen dat de krokodil de zoon zal teruggeven. In dat geval heeft de krokodil een lekker hapje voor zich.

Maar de vader denkt logisch na en zegt heel slim dat de krokodil het kind zal houden. De krokodil staat nu voor een onmogelijk dilemma:

- In het geval de krokodil beslist om het kind te houden, schendt hij zijn voorwaarden want de voorspelling is dan correct en de krokodil moet het kind teruggeven.
- Indien de krokodil echter beslist om het kind terug te geven, schendt hij nog steeds zijn voorwaarden, zelfs als deze beslissing is gebaseerd op het vorige resultaat: de voorspelling van de ouder is vals, en het kind mag niet worden teruggestuurd.

Opgave

Vraag aan de gebruiker wie stalen zenuwen heeft. De uitvoer is afhankelijk van wat de gebruiker opgegeven heeft.

- Indien vader werd opgegeven, print je: Krokodil geeft kind terug.
- Indien krokodil werd opgegeven, print je: Krokodil eet kind op.
- In alle andere geval toon je: Moe van het denken.

Invoer

```
vader
```

Uitvoer

```
Krokodil geeft kind terug
```

Invoer

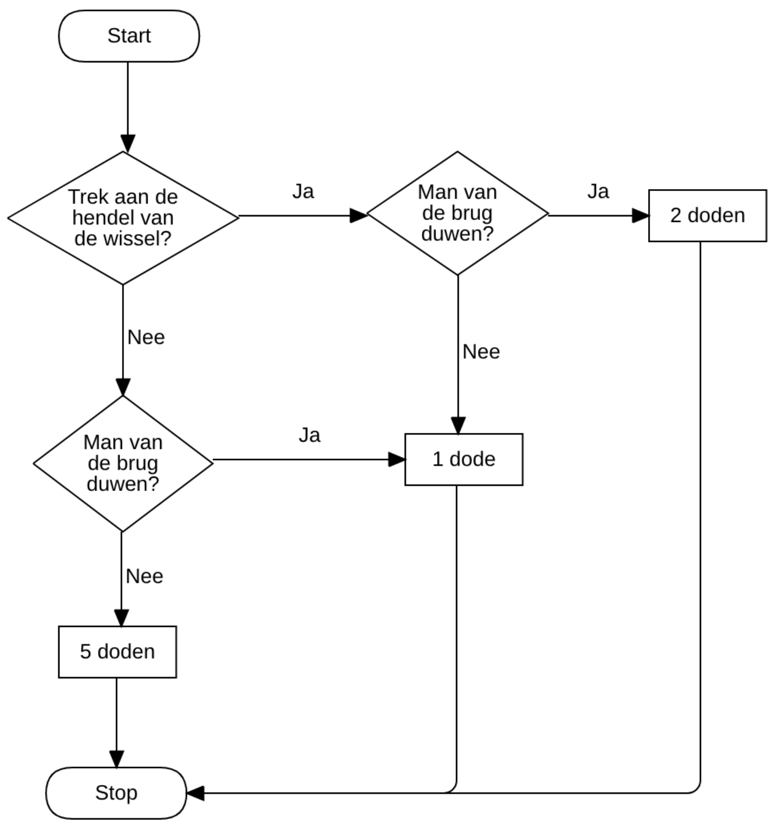
```
Tom
```

Uitvoer

```
Moe van het denken
```

6.2 Trolleyprobleem

Schrijf een programma die volgende stroomdiagram implementeert:



Figuur 5: Stroomdiagram

Invoer

```
ja
ja
```

Uitvoer

```
2
```

Invoer

```
nee
ja
```

Uitvoer

```
1
```

6.3 Schaal van Saffir-Simpson

De **schaal van Saffir-Simpson** of Saffir-Simpson hurricane schaal (SSHS) is een classificatie die in de meteorologie wordt gehanteerd om tropische cyclonen naar hun kracht in te delen. Alle tropische cyclonen zijn gevaarlijk, maar sommige zijn gevaarlijker dan andere. Daarom is er een classificatie ontwikkeld om onderscheid te maken tussen bijvoorbeeld krachtige en verwoestende orkanen en om zich beter op de te verwachten schade te kunnen voorbereiden.

De schaal wordt gebruikt om een inschatting te maken van mogelijke schade wanneer de orkaan de kust bereikt. De schaal combineert te verwachten schade aan windsnelheid enerzijds en stormvloed anderzijds. Een orkaan van categorie 2, 3, 4 en 5 is respectievelijk 10, 50, 100 en 250 maal zo verwoestend als een zwakke orkaan van categorie 1.

In de volgende tabel lees je het verband tussen windsnelheid en categorie af.

Windsnelheid (km/h)	Categorie
119-153	1
154-177	2
178-209	3
210-249	4
≥ 250	5

Opgave

Het programma leest een windsnelheid van een orkaan in. Vervolgens bepaalt het programma tot welke categorie op de schaal van Saffir-Simpson de orkaan behoort.

Invoer

```
184
```

Uitvoer

```
categorie 3
```

Invoer

```
94
```

Uitvoer

```
geen orkaan
```

6.4 Blad steen schaar

In het spelletje blad-steen-schaar verbergen twee spelers een hand achter de rug. Met hun verborgen hand vormen ze een blad, steen of schaar. Ze tonen hun hand tergelijkertijd aan elkaar. De winnaar wordt als volgt bepaald:

- Blad wint van steen.
- Steen wint van schaar.
- Schaar wint van blad.

Opgave

Je vraagt aan beide spelers de vorm die ze achter de rug verbergen. Als uitvoer toon je de speler die wint. De wedstrijd is onbeslist in het geval beide spelers dezelfde vorm verbergen.

Invoer

```
blad
blad
```

Uitvoer

```
onbeslist
```

Invoer

```
steen
blad
```

Uitvoer

```
speler 2 wint
```

6.5 Risk

Wanneer je in het spelletje **Risk** met je legers een naburig leger aanvalt volgt een spelletje met dobbelstenen tot één van de legers volledig uitgeroeid wordt. Het aantal beschikbare dobbelstenen hangt af van de rol die je speelt tijdens een aanval.

- **A**, de aanvaller, beschikt over drie dobbelstenen;
- **V**, de verdediger, beschikt over twee dobbelstenen.

Aanvallen

Nadat beide spelers hun dobbelstenen geworpen hebben, sorteren ze eerst de dobbelstenen van groot naar klein.

- **A**: ⚅ ⚁ ⚀
- **V**: ⚄ ⚂

Vervolgens vergelijken beide spelers de dobbelsteen met het hoogste aantal ogen. In dit geval wint de aanvaller want 6 is groter dan 5. De verdediger verliest één leger.

- **A**: **⚅** ⚁ ⚀
- **V**: **⚄** ⚂

Ten slotte vergelijken beide spelers de dobbelsteen met het op één na hoogste aantal ogen. In dit geval wint de verdediger want 3 is groter dan 2. De aanvaller verliest één leger.

- **A**: ⚅ **⚁** ⚀
- **V**: ⚄ **⚂**

In dit voorbeeld verliezen zowel de aanvaller als de verdediger een leger.

Speciaal geval

Er is een speciaal geval bij het vergelijken van de dobbelstenen: bij een gelijk aantal ogen verliest de aanvaller altijd. In de volgende aanval verliest de aanvaller twee legers.

- **A**: **⚄** **⚄** ⚃
- **V**: ⚄ ⚄

Opgave

Het programma leest vijf regels in:

- Eerst het aantal ogen van de drie dobbelstenen van de aanvaller, telkens op een andere regel.
- Vervolgens het aantal ogen van de twee dobbelstenen van de verdediger, ook op verschillende regels.

Let op, wanneer je het aantal ogen van de dobbelstenen opgeeft, hoeven die niet gesorteerd te zijn.

Bereken het aantal legers de aanvaller en de verdediger verliezen. Laat één van volgende regels uitschrijven:

- aanvaller verliest 0 legers, verdediger verliest 2 legers
- aanvaller verliest 2 legers, verdediger verliest 0 legers
- aanvaller verliest 1 leger, verdediger verliest 1 leger

Invoer

```
5
6
2
4
2
```

Uitvoer

```
aanvaller verliest 0 legers, verdediger verliest 2 legers
```

Invoer

```
2
4
2
1
6
```

Uitvoer

```
aanvaller verliest 1 leger, verdediger verliest 1 leger
```

7 Condities +

Oefeningen	Moeilijksheidsgraad
Atoommodel van Bohr	★★★
Codontype	★★★
Hertzsprung-Russeldiagram	★★★
Monsters en hoeden	★★★

7.1 Atoommodel van Bohr

Schillen worden volgens toenemende afstand tot de kern voorgesteld door de letters K, L, M, N, O, P en Q. Het rangnummer wordt het schilnummer n genoemd. Een schil met rangnummer n kan maximaal $2n^2$ elektronen bevatten, zoals aangegeven in de onderstaande tabel (deze regel is geldig voor $n = 1$ tot en met $n = 4$, terwijl $n = 5, 6, 7$ telkens maximaal 32 elektronen heeft):

Schil	K	L	M	N	O	P	Q
Nummer n	1	2	3	4	5	6	7
Maximale bezetting ($\leq 2n^2$)	2	8	18	32	32	32	32
Cumulatieve bezetting	2	10	28	60	92	124	156

Gegeven het aantal elektronen van een atoom dat zich in stabiele toestand bevindt. Wat is dan de buitenste gevulde schil? In stabiele toestand worden de schillen van binnen naar buiten opgevuld. Een natriumatoom heeft bijvoorbeeld elf elektronen. In de stabiele toestand zitter er dus 2 elektronen in de K-schil, 8 in de L-schil en het laatste elektron zit in de M-schil.

De invoer bevat het aantal elektronen e in een atoom, waarbij $1 \leq e \leq 156$. De uitvoer bevat de van de buitenste schil als het atoom zich in stabiele toestand bevindt, weergegeven op basis van de volgende template: `"De *M*-schil is de buitenste schil van een stabiel atoom met *11* elektronen."`. De cursieve fragmenten moeten uiteraard correct ingevuld worden op basis van de gegevens uit de invoer.

Invoer

```
11
```

Uitvoer

```
De M-schil is de buitenste schil van een stabiel atoom met
11 elektronen.
```

7.2 Codontype

De genetische code is een verzameling regels die aangeven hoe levende cellen de informatie die zit opgeslagen in hun genetisch materiaal (DNA of RNA sequenties) vertalen naar eiwitten (aminozuursequenties). Meer in het bijzonder definieert deze code een vertaling van trinucleotidesequenties (codons) naar aminozuren, waarbij elk triplet van nucleotiden in een nucleotidesequentie correspondeert met één enkel aminozuur. Bij de vertaling van RNA naar eiwitten kunnen we drie types van codons onderscheiden: de vertaling begint bij een startcodon (AUG) en loopt verder tot één van de stopcodons (UAG (amber), UGA (opaal) en UAA (oker)) bereikt wordt. Daartussen bevinden zich een willekeurig aantal gewone codons, zoals geïllustreerd in onderstaande figuur.

De invoer bevat één enkele regel tekst die een RNA codon moet voorstellen (dus een string die enkel bestaat uit de karakters A, C, G en U). De uitvoer bevat één enkele regel met de omschrijving van het type van het codon uit de invoer. De vier mogelijke types met bijhorende omschrijvingen staan weergegeven in onderstaande tabel.

Type	omschrijving
startcodon (AUG)	start
stopcodon (UAG, UGA, UAA)	stop
gewoon codon (elk ander triplet van nucleotiden)	gewoon
ongeldig codon (string met lengte verschillend van drie)	ongeldig

Als formaat voor de uitvoer gebruik je de volgende template: "Het codon GCC is een gewoon codon.". De cursieve fragmenten van deze template zijn variabel, en moeten ingevuld worden op basis van de gegeven invoer en de berekende omschrijving van het codontype. Probeer het aantal voorwaarden en de uitdrukking van de voorwaarden die moeten getest worden om de uitvoer te bepalen zo beknopt mogelijk te houden. Het is niet nodig om na te gaan of de gegeven invoer enkel basen uit het RNA alfabet (A, C, G en U) bevat om een geldig codon (gewoon, start of stop) voor te stellen.

Invoer

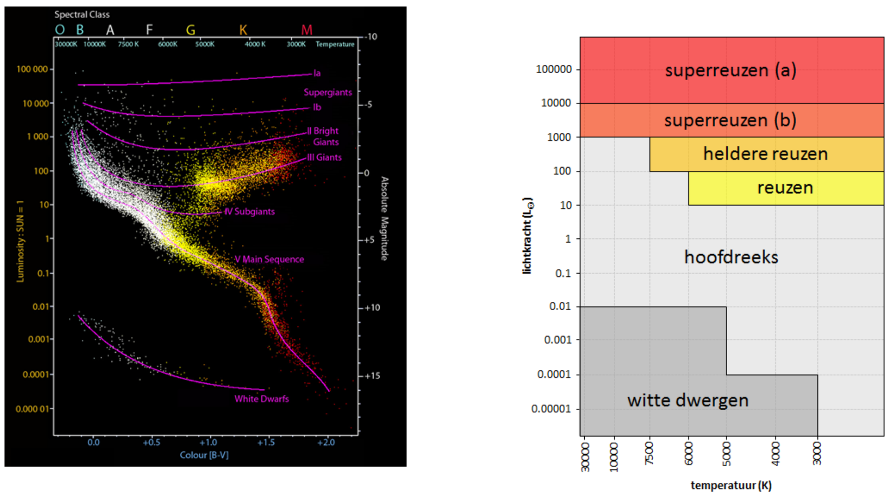
```
GCC
```

Uitvoer

```
Het codon GCC is een gewoon codon.
```

7.3 Hertzsprung-Russelldiagram

Op basis van de positie van de sterren binnen een Hertzsprung-Russelldiagram worden de volgende klassen onderscheiden: superreuzen (a), superreuzen (b), heldere reuzen, reuzen, de hoofdreeks en witte dwergen. Er bestaat geen eenduidige afbakening van de gebieden in het diagram die corresponderen met de verschillende klassen. Voor deze opgave zullen we uitgaan van de afbakening die staat weergegeven in de rechter figuur hierboven. De randen van elk gebied liggen daarbij telkens parallel met één van de assen van het diagram en vallen steeds samen met één van de waarden die op de assen staan weergegeven.



Figuur 6: Hertzsprung-Russelldiagram

De invoer bestaat uit twee regels die elk een floating point getal bevatten, die respectievelijk de temperatuur (in Kelvin) en de lichtkracht (relatief ten opzichte van de zon) van een ster voorstellen. Je mag ervan uitgaan dat de waarden van de temperatuur en de lichtkracht steeds binnen de grenzen van de rechter figuur hierboven vallen, en dat ze nooit een punt vormen dat op de rand ligt van de gebieden die bij de indeling gebruikt worden. De uitvoer bevat een omschrijving die aangeeft tot welke klasse de ster behoort, volgens de indeling die gemaakt wordt in de rechter figuur hierboven.

Invoer

```
8525.0
196000.0
```

Uitvoer

```
superreuzen (a)
```

7.4 Monsters en hoeden

Er bestaat een onfeilbare strategie die garandeert dat er altijd juist één persoon de juiste kleur als antwoord geeft. De makkelijkste manier om deze strategie uit te leggen is door vast te stellen dat de twee hoeden ofwel **dezelfde** kleur of een **verschillende** kleur hebben. Als je dat eenmaal doorhebt, dan bestaat de strategie erin te beslissen wie van beide personen **hetzelfde** zal zeggen en wie **verschillend** zal zeggen op basis van de kleur van de hoed die ze te zien krijgen. Met andere woorden, de personen moeten afspreken dat één persoon altijd de kleur van de hoed zal zeggen die hij of zij ziet, en dat de andere persoon altijd de omgekeerde kleur zal zeggen die hij of zij ziet. Het lijkt misschien onwaarschijnlijk, maar uit onderstaande afbeelding blijkt dat deze strategie er altijd toe zal leiden dat juist één van de twee personen het juiste antwoord zal geven, en hen daardoor redt om door het monster verslonden te worden.

Figuur 7: Illustratie monsters en hoeden

De invoer bevat 3 regels met de eerste twee regels telkens de kleur van de hoed die op het hoofd staat van respectievelijk de eerste en de tweede persoon: zwart of wit. De derde regel bevat een getal (1 of 2) dat aangeeft wie van de twee personen de omgekeerde kleur zal zeggen. De andere persoon zegt dan dezelfde kleur. De uitvoer bestaat uit twee regels, die de kleur bevatten (zwart of wit) die als antwoord gegeven wordt door respectievelijk de eerste en de tweede persoon, op basis van de strategie zoals hierboven uitgelegd.

Invoer

```
wit
zwart
2
```

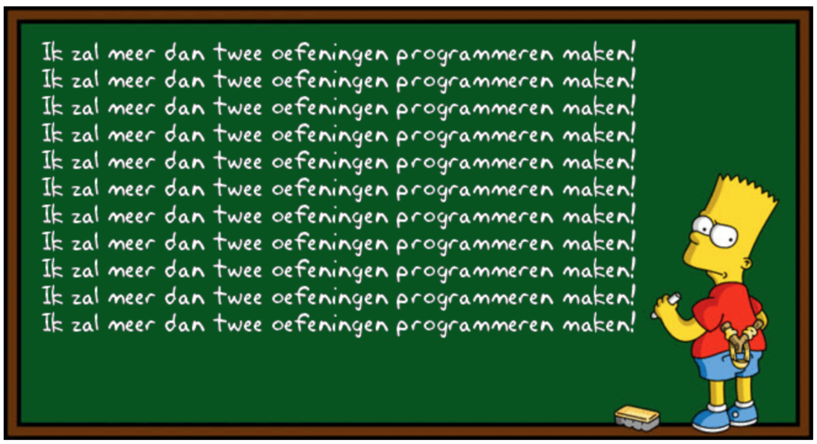
Uitvoer

```
zwart
zwart
```

8 Iteraties a

Oefeningen	Moeilijkheidsgraad
Bart Simpson	☆☆☆
Omkeren	★☆☆
Veelvouden	★☆☆
Het grootste en het gemiddelde	★★☆
De rij van Fibonacci	★★☆

8.1 Bart Simpson



Figuur 8: Bart Simpson

Opgave

Schrijf een programma die de tekst op het bord uitschrijft.

Uitvoer

```
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
Ik zal meer dan twee oefeningen programmeren maken!
```

8.2 Omkeren

Schrijf een programma dat een woord inleest en het omgekeerde woord print.

Invoer

```
programmeren
```

Uitvoer

```
neremmargorp
```

8.3 Veelvouden

Gegeven is getal $r \in \mathbb{N}_0$ met $r < 100$. Bepaal de som van alle veelvouden van r die kleiner dan of gelijk zijn aan 100.

Invoer

```
45
```

Uitvoer

```
135
```

8.4 Het grootste en het gemiddelde

Schrijf een programma dat aan de gebruiker minstens 2 getallen $x_i \in \mathbb{Z}$ vraagt. Het programma toont:

- het grootste van de getallen;
- het gemiddelde van de getallen.

De wiskundige formule om het gemiddelde $\overline{x}$ van n getallen te berekenen is:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (n \in \mathbb{N}_0^+)$$

Opgave

Op de eerste regel van de invoer staat hoeveel getallen je inleest. Dit getal is steeds minstens 2. Vervolgens staan de getallen waarvan je het grootste en het gemiddelde berekent. In de uitvoer rond je het gemiddelde af tot op 2 cijfers na de komma.

Invoer

```
4
6
-22
26
46
```

Uitvoer

```
Het grootste getal is 46 en het gemiddelde is 14.00
```

8.5 De rij van Fibonacci

De rij van Fibonacci is een rij getallen waarvan de eerste twee getallen gelijk zijn aan 1. Elk volgend getal is de som van de twee voorgaande getallen. De eerste elementen van de rij zijn dan als volgt:

$$1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

De manier waarop de rij van Fibonacci gedefinieerd is, is een voorbeeld van wat in de wiskunde een recursieve definitie genoemd wordt:

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_n = F_{n-1} + F_{n-2} \quad \text{voor } n > 2 \end{cases}$$

Opgave

De invoer bestaat uit een getal $n \in \mathbb{N}_0$. Schrijf het n-de getal in de rij van Fibonacci uit.

Invoer

```
3
```

Uitvoer

```
2
```

Invoer

```
101
```

Uitvoer

```
573147844013817084101
```

8.6 Iteraties a+

Oefeningen	Moeilijksheidsgraad
Chaos	★★★
Lifters	★★★
Eerste verwittiging	★★★

8.7 Chaos

Het volgende eenvoudige model voor de groei van een populatie kan bijvoorbeeld toegepast worden op het visbestand in een meer, bacteriën in een proefbuis, of tal van andere gelijkaardige omstandigheden. Hierbij wordt verondersteld dat een populatie kan schommelen tussen 0 (volledig uitgestorven) en 1 (maximale populatie die door de omgeving getolereerd wordt).

Als de grootte of de dichtheid van de populatie op tijdstip t_i gelijk is aan d, dan veronderstellen we dat de populatie op tijdstip t_{i+1} gelijk is aan $rd(1-d)$, waarbij het argument r — *de vruchtbaarheidsparameter* genoemd — de mate van groei bepaalt.

De invoer bevat drie regels die achtereenvolgens de volgende informatie bevatten:

- de initiële populatiedichtheid $d \in \mathbb{R}$;
- de waarde van de vruchtbaarheidsparameter $r \in \mathbb{R}$;
- het aantal tijdsstappen $s \in \mathbb{N}_0$ waarover we de populatiedichtheid willen simuleren (inclusief het tijdstip t_0).

De uitvoer bevat s regels die de populatiedichtheid bevatten op de tijdstippen $t_0, t_1, \ldots, t_{s-1}$.

Als r rond de **2** ligt, dan bendadert de populatiedichteid na verloop van tijd de waarde $1 - \frac{1}{r}$

Invoer

```
0.1
1.9
6
```

Uitvoer

```
0.1
0.171
0.26934210000000003
0.373914173018421
0.44479449204530547
0.4692094685937818
```

Als $r = \mathbf{3}$ convergeert de populatiedichtheid naar een toestand waarbij alternerend tusssen twee waarden gesprongen wordt.

Invoer

```
0.1
3
12
```

Uitvoer

```
0.1
0.2700000000000001
0.5913000000000002
0.7249929299999999
0.5981345443500454
0.7211088336156269
0.603332651091411
0.7179670896552621
0.6074710434816448
0.7153499244388992
0.6108732301324811
0.7131213805199696
```

Als $r \geq \mathbf{4}$ dan wordt het gedrag van het model chaotisch. Dat betekent dat de resultaten extreem kunnen verschillen bij zeer kleine schommelingen van de begintoestand. Vergelijk bijvoorbeeld het verschil in gedrag tussen dit voorbeeld en het volgende voorbeeld.

Invoer

```
0.1
4
60
```

Uitvoer

```
0.1
0.36000000000000004
0.9216
...  (54 regels) ...
0.977464119602946
0.08811205796713474
0.32139329283172413
```

Een verschil van 0.00000000001 in de initiële populatiedichtheid resulteert in een groot verschil in de populatiedichtheid op tijdstip t_{59}. Dit is chaotisch gedrag, en het voorbeeld toont aan dat de populatiedichtheid extreem moelijk te voorspellen is, omdat kleine verschillen bij aanvang op termijn kunnen resulteren in grote verschillen.

Invoer

```
0.10000000001
4
60
```

Uitvoer

```
0.10000000001
0.36000000003200006
0.9216000000358401
...  (54 regels) ...
0.830632498181969
0.5627286045838009
0.9842604886678766
```

8.8 Lifters

Je vertrekt met de auto naar het zuiden van Frankrijk, en je weet dat je onderweg n lifters zal tegenkomen. Omdat je graag wat gezelschap wil tijdens de lange autorit, beslis je om één van deze lifters mee te nemen. Als je dan toch mag kiezen, dan zou je er liefst ook de knapste (of sympatiekste, slimste, rijkste, ...) uitpikken. Terugkeren is echter uitgesloten, dus moet je bij elke lifter onmiddellijk beslissen of je hem/haar zult meenemen dan wel of je het erop zult wagen om verder nog een knappere lifter tegen te komen.

Voor je vertrek heb je de volgende strategie uitgedokterd om met grote kans de knapste lifter op te pikken. Telkens je een lifter tegenkomt, geef je hem/haar ogenblikkelijk een score $s \in \mathbb{R}$ $(0 \leq s \leq 1)$ die aangeeft hoe knap de lifter is. Hoe hoger de score, hoe knapper.

De eerste $\frac{n}{2}$ (gehele deling) lifters neem je niet mee, maar je onthoudt wel de hoogste score die je aan die lifters hebt gegeven. Daarna neem je de eerste lifter mee die een hogere score heeft dan de hoogste score van de eerste $\frac{n}{2}$ lifters. Je neemt de laatste lifter mee, als je daarvoor nog geen lifter hebt meegenomen.

Ter info

Het is makkelijk in te zien dat bovenstaande strategie je 25% kans geeft om de knapste lifter mee te nemen. Dat laatste zal immers het geval zijn als de tweede beste lifter zich in de eerste helft van de lifters bevindt, en de beste lifter in de tweede helft.

Het is zelfs mogelijk om de kans nog lichtjes te verbeteren tot $\frac{1}{e} = 0.36788$ door in grote lijnen dezelfde strategie te volgen, maar door vanaf lifter $\frac{n}{e}$ te beginnen beslissen of je hem/haar meeneemt of niet.

De invoer bevat in de eerste regel van de invoer een getal $n \in \mathbb{N}_0$ dat aangeeft hoeveel lifters er zijn. Daarna volgen de scores $s \in \mathbb{R}$ $(0 \leq s \leq 1)$ die aan de n lifters gegeven worden, elk op een afzonderlijke regel. De uitvoer bevat de score s van de lifter die uiteindelijk wordt meegenomen als de hierboven omschreven procedure gevolgd wordt.

Invoer

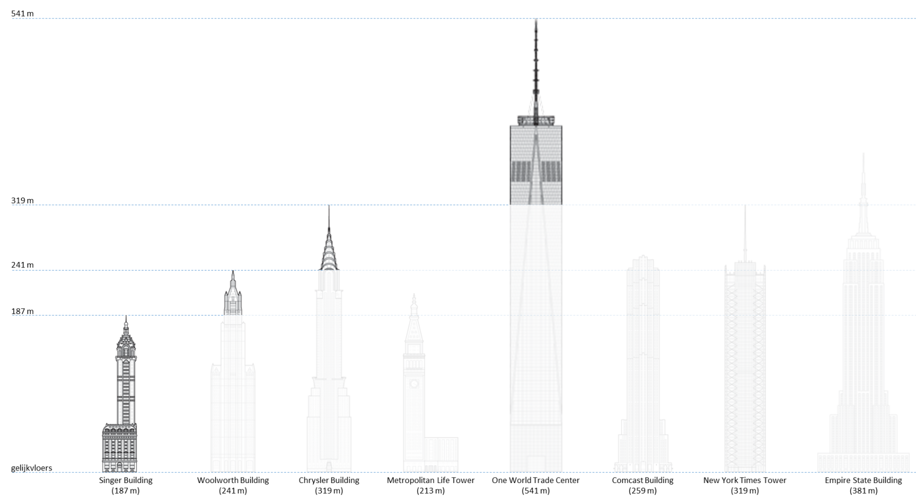
```
19
0.2583
0.1580
0.4293
0.2299
0.2443
0.1043
0.0632
0.0363
0.1381
0.9899
0.3766
0.7932
0.7567
0.1048
0.9148
0.3787
0.7712
0.1390
0.4001
```

Uitvoer

```
0.9899
```

8.9 Eerste verwittiging

We beschouwen de skyline van een stad met een aantal grote gebouwen die allemaal naast elkaar staan. Hieronder zie je bijvoorbeeld de skyline van New York City waarin een aantal van de grootste gebouwen op schaal weergegeven worden.



Figuur 9: De skyline van New York City op schaal

Als je deze gebouwen vanaf een grote afstand aan de linkerkant bekijkt, welke stukken van welk gebouw zijn dan zichtbaar als we ervan uitgaan dat achterliggende gebouwen volledig aan het zicht onttrokken worden door de gebouwen die er voor staan.

De invoer bevat de eerste regel een getal n $\mathbb{N}_0$ dat aangeeft hoeveel gebouwen er naast elkaar staan. Daarna worden de n gebouwen van links naar rechts omschreven door hun naam en hun hoogte in meter (een natuurlijk getal), elk op een afzonderlijke regel. In de uitvoer omschrijf je de stukken van de gebouwen die zichtbaar zijn, als je de gegeven skyline vanaf een grote afstand aan de linkerkant bekijkt. Hierbij mag je ervan uitgaan dat achterliggende gebouwen volledig aan het zicht onttrokken worden door de gebouwen die er voor staan. De (stukken van) gebouwen moeten opgelijst worden volgens stijgende afstand van de gebouwen tot het gezichtspunt. Het exacte formaat van de uitvoer kan je afleiden uit onderstaand voorbeeld. Dit voorbeeld komt trouwens overeen met de skyline in bovenstaande figuur.

Invoer

```
8
Singer Building
187
Woolworth Building
241
Chrysler Building
319
Metropolitan Life Tower
213
One World Trade Center
541
Comcast Building
259
New York Times Tower
319
Empire State Building
381
```

Uitvoer

```
Singer Building is zichtbaar van het gelijkvloers tot 187
meter.
Woolworth Building is zichtbaar van 187 meter tot 241
meter.
Chrysler Building is zichtbaar van 241 meter tot 319 meter.
One World Trade Center is zichtbaar van 319 meter tot 541
meter.
```