

# De eerste ronde Nederlandse Informatica Olympiade 2021-2022

---



informatica  
olympiade

De informatica olympiade is een wedstrijd voor leerlingen uit het voortgezet onderwijs in Nederland. Het is een wedstrijd die bestaat uit drie ronden. In de derde ronde wordt bepaald wie Nederland mogen vertegenwoordigen op de Internationale Informatica Olympiade in zomer 2022 in Indonesië.

## De eerste ronde

De eerste ronde van de Nederlandse Informatica Olympiade bestaat dit jaar uit 12 opgaven. Die hoeft je niet allemaal te maken, al mag dat natuurlijk wel. Deelnemers die tenminste 200 punten halen krijgen een certificaat.

Heb je tussen de 200 en 399 punten dan staat op het Certificaat de vermelding **Brons**, tussen de 400 en 599 punten de vermelding **Zilver** en bij 600 punten of meer punten de vermelding **Goud**.

Soort	Omschrijving	Aantal	Punten per opgave	Totaal te behalen
A	Inleidende opgaven	5	40	200
B	Theoretische opgaven	4	50	200
C	Gevorderde opgaven	2	100	200
D	Een spel programmeren	1	100	100

De beste 100 leerlingen worden uitgenodigd voor de tweede ronde, die in maart 2022 wordt gehouden op de Universiteit Twente. Voor deelname aan de tweede ronde moet je wel minstens 200 punten hebben gehaald.

Voor de beste deelnemer van iedere klas is een aparte prijs beschikbaar.

Om deel te kunnen nemen moet je een account maken op [submit.informaticaolympiade.nl](https://submit.informaticaolympiade.nl)

Bij de eerste keer aanmelden moet je enkele gegevens aanleveren die wij nodig hebben om de olympiade goed te kunnen organiseren. Als je deze gegevens niet wilt of kunt aanleveren, kun je helaas niet deelnemen. Je verklaart in de laatste stap dat je de gegevens naar waarheid hebt ingevuld; daarna staat deelname voor je open. Als je van vorige jaren al een account hebt, zul je de gegevens ook eventueel eerst moeten aanvullen voor je verder kunt werken in het systeem. Wij gaan

zeer zorgvuldig om met de gegevens die je ons aanlevert. Wij zullen deze gegevens niet met derden delen.

Je kunt je uitwerkingen uploaden naar [submit.informaticaolympiade.nl](https://submit.informaticaolympiade.nl) wanneer je in het systeem bent ingelogd. In het systeem kun je ook een voorbeeldopgave insturen om uit te proberen hoe het werkt. De opgaven worden meteen geheel of gedeeltelijk nagekeken, voor de rest van de uitslag zul je moeten wachten op het resultaat. Je uitwerkingen voor de opgaven A, B en C moeten uiterlijk 19 januari worden geüpload. Op 22 januari wordt de eerste ronde gejureerd en kort daarna worden de uitslagen gepubliceerd.

Voor de spelopgave, opgave D, moet je je aanmelden op [www.codecup.nl](https://www.codecup.nl) en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 22 januari 2022 tegen elkaar uit in een toernooi dat te volgen is op [www.codecup.nl](https://www.codecup.nl). De beste leerling of docent uit het voorgezet onderwijs wint de jaarlijkse Windesheim Digitalisprijs van 200 euro. Inzenden mag tot 22 januari 7.00 u. Dan begint het toernooi.

Voor alle opgaven geldt dat je ervan uit mag gaan dat je programma's alleen correcte invoer aangeboden krijgen.

#### Opgaven A1 tot en met A5

Deze opgaven zijn vooral bedoeld voor leerlingen die beginnen met programmeren. Vanuit de olympiade bieden we lesmateriaal aan om te beginnen met programmeren met Python. Dat is de cursus CS Circles van de Universiteit van Waterloo in Canada. Er is een Nederlandse vertaling beschikbaar op [cursus.informaticaolympiade.nl](https://cursus.informaticaolympiade.nl). In de tekst van die Nederlandse vertaling staat aangegeven wanneer je toe bent aan de volgende opgave van de eerste ronde.

#### Opgaven B1 tot en met B4

Deze opgave kun je één voor één downloaden uit het inzendsysteem. De opgave wordt speciaal voor jou gemaakt en jij moet het antwoord op de opgave die je vanuit het systeem krijgt inleveren. Het heeft dus geen zin om de antwoorden van iemand anders te gebruiken en die in te zenden.

Als je binnen een week na downloaden het goede antwoord instuurt krijg je 50 punten per opgave. Voor iedere dag later gaat er één punt van je score af. Inzendingen na 19 januari 2022 zullen niet worden verwerkt.

Als je een verkeerd antwoord hebt gegeven, verlies je meteen 10 punten, totdat er van de 50 punten geen punten meer over zijn.

Het gaat bij al deze opgaven om korte antwoorden, een getal of een korte tekst, die je op de betreffende pagina van het inzendsysteem kunt invoeren. Als je je antwoord hebt bevestigd, krijg je meteen je score te zien.

Je mag allerlei hulpmiddelen gebruiken om de opgave op te lossen. Je zou er bijvoorbeeld een computerprogramma bij kunnen schrijven. Noodzakelijk is dat echter niet. Als voorbereiding op het vervolg van de informatica olympiade is het wel een mooie uitdaging om na te gaan hoe je een programma zou kunnen schrijven dat dit probleem, of problemen die er op lijken, kunt oplossen.

## Opgaven C1 en C2

Dit zijn wat complexere opgaven waarmee je een probleem moet oplossen door het schrijven van een computerprogramma. Die programma's lezen invoer van standard input (het toetsenbord) en schrijven uitvoer naar standard output (het beeldscherm). Je programma moet zich daarbij precies houden aan de beschrijvingen van de opdracht. Je programma krijgt een aantal testgevallen voorgeschoteld en voor ieder testgeval kun je punten krijgen.

## Opgave D en de CodeCup

Bij deze opgave moet je een programma schrijven dat het spel Spaghetti kan spelen. Aan dit toernooi doen ook andere deelnemers me, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 22 januari een toernooi tegen elkaar. Om deel te kunnen nemen moet je programma kunnen samenwerken met onze jurysoftware; voor details verwijzen we naar [www.codecup.nl](http://www.codecup.nl)

In principe wordt je deelname door ons opgemerkt en wordt je aangemerkt als een NIO-deelnemer. Je kan ook even een e-mail sturen aan [contest@codecup.nl](mailto:contest@codecup.nl) en dan zorgen wij ervoor dat je een NIO-deelnemer wordt. Alle punten die je haalt in de CodeCup komen automatisch ook bij de puntentelling van [submit.informaticaolympiade.nl](http://submit.informaticaolympiade.nl) terecht.

## **European Girls' Olympiad in Informatics (EGOI)**

Ben je een meisje en wil je kans maken om Nederland bij de European Girls' Olympiad in Informatics (EGOI) te vertegenwoordigen?

Geef je dan op via [girls@informaticaolympiade.nl](mailto:girls@informaticaolympiade.nl)

We organiseren ook een programmeercursus als voorbereiding op de EGOI. De cursus start na de herfstvakantie.

De beste vier deelnemers aan de NIO en de cursus mogen deelnemen aan de EGOI in juni 2022 in Turkije.

Aan zowel de cursus als de eventuele deelname aan de EGOI zijn geen kosten verbonden.

.

# Opgave A1. Verschillende letters

---

Schrijf een programma dat een woord inleest van een regel van standard input. Het woord bestaat uit minstens 1, hoogstens 30 letters, geschreven in kleine letters, zonder leestekens of spaties.

Het programma schrijft naar standard output een regel met daarop het aantal verschillende letters in dat woord.

Voorbeeld

Invoer:        aardappelsalade

Uitvoer:       7

Toelichting:

De letters in het woord zijn de *a, d, e, l, p, r, s*.

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave A2. AB-code

---

Bij elk positief geheel getal  $n$  kun je een unieke code bepalen die bestaat uit twee getallen  $m$  en  $p$ . Dat werkt als volgt:

Stap 1: Schrijf  $n$  als een getal in het tweetalig stelsel, als een binair getal  $b$ . Zie voor toelichting hieronder.

Stap 2: Schrijf  $b$  achterstevoren als een binair getal  $a$ .

Stap 3: Het getal  $m$  is het aantal nullen waarmee getal  $a$  begint.

Stap 4: Het getal  $p$  krijg je door het binaire getal  $a$  om te schrijven als getal in het tientalig stelsel.

Stel bijvoorbeeld  $n = 202$ .

1. Binair kun je dat schrijven als 11001010.
2. Als je dat achterstevoren schrijft krijg je 01010011.
3. Dat getal begint met 1 keer een 0.
4. Als je die weglaat krijg je 1010011. Schrijf dat weer in het tientalig stelsel en je krijgt 83.

De AB-code van 202 is dan het paar 1 en 83.

Schrijf een programma dat een getal  $n$  inleest van standard input. Er geldt  $0 < n < 1000000000$ .

Het programma schrijft twee regels naar standard output. Op de eerste regel staat het getal  $m$  en op de tweede het getal  $p$ .

Voorbeeld:

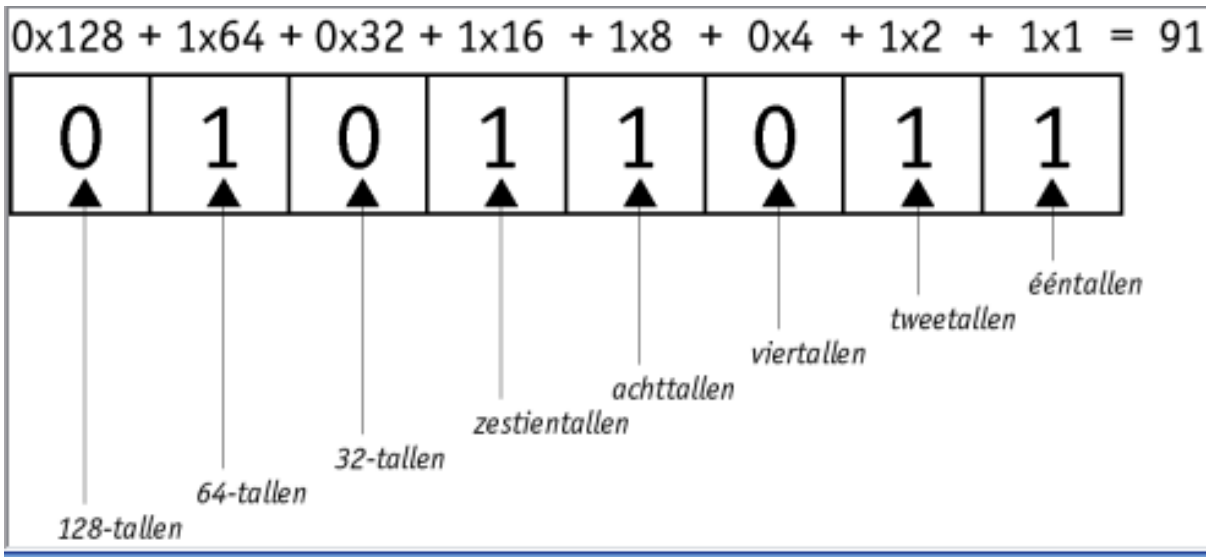
Invoer: 202

Uitvoer: 1  
83

Voor je programma geldt een tijdslimiet van 2 seconden.

## Binaire schrijfwijze van getallen

Binaire getallen zijn getallen die met grondtal 2 zijn geschreven. Uitsluitend de symbolen 0 en 1 worden gebruikt. Van achteren naar voren krijg je eerst 0 of 1 keer  $2^0$ , daarvoor 0 of 1 keer  $2^1$ , dan 0 of 1 keer  $2^2$  etc.



# Opgave A3. Gelukkige getallen?

---

Sommige positief gehele getallen zijn een gelukkig getal. Hoe kun je dat nagaan?

Herhaal het volgende: bepaal van het getal de som van de kwadraten van de cijfers. En doe hetzelfde met het resultaat. Kom je op een 1 uit, dan was het getal waarmee je bent begonnen een gelukkig getal. Kom je in een lusje, bij een getal wat je al eerder hebt gehad, dan is het getal waarmee je begon geen gelukkig getal.

Voorbeeld 1:

85	gaat naar $64 + 25 = 89$
89	gaat naar $64 + 81 = 145$
145	gaat naar $1 + 16 + 25 = 42$
42	gaat naar $16 + 4 = 20$
20	gaat naar 4
4	gaat naar 16
16	gaat naar $1 + 36 = 37$
37	gaat naar $9 + 49 = 58$
58	gaat naar $25 + 64 = 89$
89	hebben we al eerder gehad

Conclusie: 85 is geen gelukkig getal

Voorbeeld 2:

82	gaat naar $4 + 64 = 68$
68	gaat naar $36 + 64 = 100$
100	gaat naar 1

Conclusie: 82 is een gelukkig getal.

Je programma leest van standard input twee regels. Op de eerste regel staat een positief geheel getal  $N$  (met  $1 \leq N \leq 10000$ ) en op de tweede regel een getal  $M$  (met  $N \leq M \leq 10000$ ).

Je programma schrijft naar standaard output één regel, met daarop het aantal gelukkige getallen  $G$  waarvoor geldt  $N \leq G \leq M$ .

Voorbeeld

Invoer:        82  
                  85

Uitvoer:       1

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave A4. Regenwormen glazen bol

---



Bij het spelletje regenwormen gooien deelnemers met acht dobbelstenen om zoveel mogelijk punten bij elkaar te verzamelen. Op elke dobbelsteen staan de cijfers 1, 2, 3, 4 en 5 en een regenwormpje.

- De speler gooit eerst met alle acht dobbelstenen.
- De speler kiest een van de geworpen symbolen (een getal of de regenworm) en legt alle dobbelstenen met dit symbool opzij.
- De speler mag het gooien en opzijleggen een aantal keren herhalen, telkens met de overgebleven dobbelstenen. Na elke worp moeten er dobbelstenen opzijgelegd worden om nogmaals te mogen gooien, maar de speler moet wel steeds voor een ander symbool kiezen: een symbool dat eenmaal opzijgelegd is, mag in dezelfde beurt niet nogmaals opzijgelegd worden.
- Als er bij een worp geen stenen opzij kunnen worden gelegd eindigt de beurt en is het resultaat van de beurt 0 punten.
- Als er geen enkel regenwormpje opzij is gelegd, is het resultaat van de beurt ook 0 punten.
- In alle andere gevallen wordt het aantal ogen van de opzij gelegde dobbelstenen bij elkaar opgeteld. Elk regenwormpje geldt voor 5 punten.

Schrijf een programma dat van standaard input één regel met 33 tekens inleest, de (mogelijke) dobbelsteenworpen tijdens één beurt bij het spel regenwormen. Ieder teken geeft aan wat er op de dobbelsteen boven komt, met een 1, 2, 3, 4, 5 of een w.

Je programma schrijft naar één regel van standard output de maximale score die met deze worpen kan worden bereikt.

Voorbeeld

Invoer:           w132243535521ww41w552311w42341w52

Uitvoer:           28



Toelichting:

Worp	Laten liggen	Puntentotaal	Nog te gooien
w1322435	4	0 (4)	7
35521ww	ww	14	5
41w55	55	24	3
231	3	27	2
1w	1	28	1

Na de eerste beurt heb je 4 punten opzij gelegd. Omdat er nog geen regenworm opzij is gelegd is het dan nog niet zeker of je de 4 punten ook werkelijk krijgt.

Met de laatste dobbelsteen zou je een 4 hebben gegooid, maar die mag je niet meer opzij leggen. Vandaar dat je past en 28 punten kunt noteren.

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave A5. Metrotrip

De gemeente Lintstad heeft een nieuwe metro laten aanleggen. De stations liggen op één lijn en zijn opvolgend genummerd van 1 tot en met N.

Elke trein doet dienst op een deel van het hele netwerk. Elke trein gaat heen en weer tussen stations  $i$  en  $j$ , in de tekst weergegeven als de trein  $(i, j)$  waarbij geldt dat  $1 \leq i < j \leq N$ .

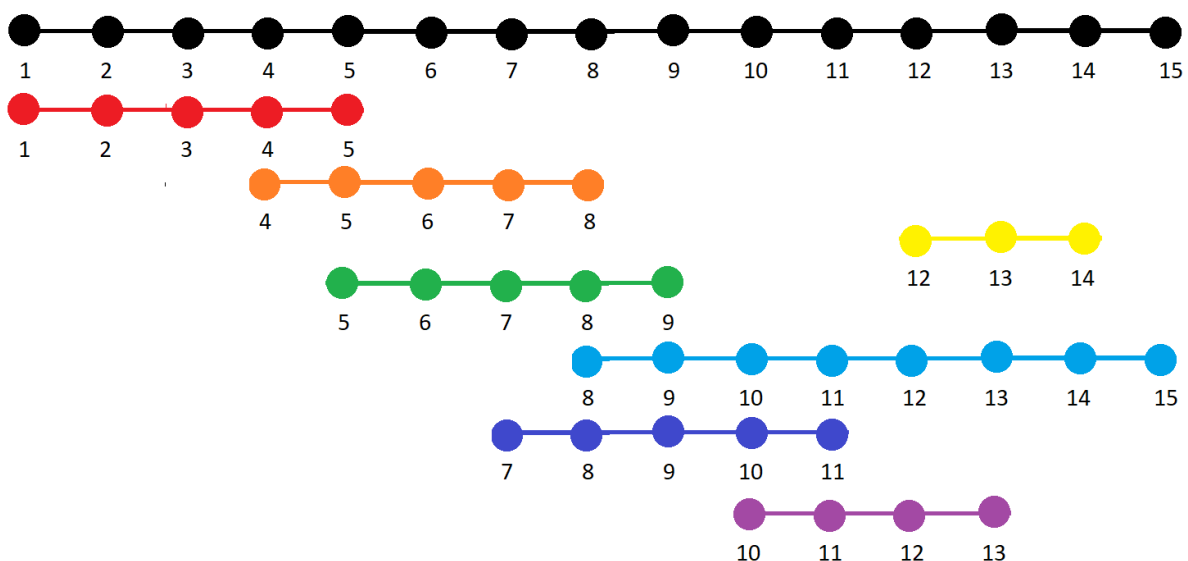
Een kaartje geeft je het recht om over te stappen. Je kunt dan in een station je trein verlaten en met een andere trein verder reizen. Er zijn twee voorwaarden aan een overstap: je moet de reis in dezelfde richting vervolgen en de trein waarmee je je reis vervolgt moet op een later station zijn vertrokken dan de trein waaruit je bent uitgestapt.

Als je een kaartje koopt bij station 1, met hoeveel verschillende treinen kun je dan maximaal reizen in de richting van station N?

Stel dat er 15 stations zijn en zeven treindiensten,  $(1, 5)$ ,  $(4, 8)$ ,  $(12, 14)$ ,  $(5, 9)$ ,  $(8, 15)$ ,  $(7, 11)$  en  $(10, 13)$ . Je kunt dan met één kaartje vanaf station 1 alle zeven treinen gebruiken.

- Bij station 1 stap je in lijn  $(1, 5)$ .
- Bij station 4 stap je over op lijn  $(4, 8)$ .
- Bij station 5 stap je over op lijn  $(5, 9)$ .
- Bij station 7 of 8 stap je over op de lijn  $(7, 11)$ .
- Bij station 8, 9 of 10 stap je over op de lijn  $(8, 15)$ .
- Bij station 10 of 11 stap je over op de lijn  $(10, 13)$ .
- Bij station 12 of 13 stap je over op de lijn  $(12, 14)$ .

Je reis eindigt dan op station 14. Let op dat je niet mag overstappen op station 14 naar de lijn  $(8, 15)$ . Die is namelijk op niet op een later station begonnen dan de lijn waar je uit bent gestapt.



Schrijf een programma dat van standard input achtereenvolgens de volgende zaken inleest.

Op de eerste regel staan twee getallen,  $N$  en  $M$ , gescheiden door een spatie. Er zijn  $N$  stations en  $M$  treinen.  $N$  is niet groter dan 10000 en  $M$  is maximaal 4000.

Op de volgende  $M$  regels staan de verschillende treinen, telkens eerst het beginstation  $I$ , dan het eindstation  $J$ , gescheiden door een spatie. Er geldt altijd  $I < J$ .

Je programma schrijft één regel naar standard output met daarop het aantal verschillende treinen dat kan worden genomen met één kaartje vanaf station 1.

Voorbeeld:

```
Invoer:      15 7
              1 5
              4 8
              12 14
              5 9
              8 15
              7 11
              10 13
```

```
Uitvoer:     7
```

Randvoorwaarden:

Voor de helft van de testgevallen geldt  $N, M \leq 100$

Voor deze opgave geldt een tijdlimiet van 2 seconden.

# Opgave B1 tot en met B4:

---

Download deze van [submit.informaticaolympiade.nl](http://submit.informaticaolympiade.nl)

B1. Max-som-subarray

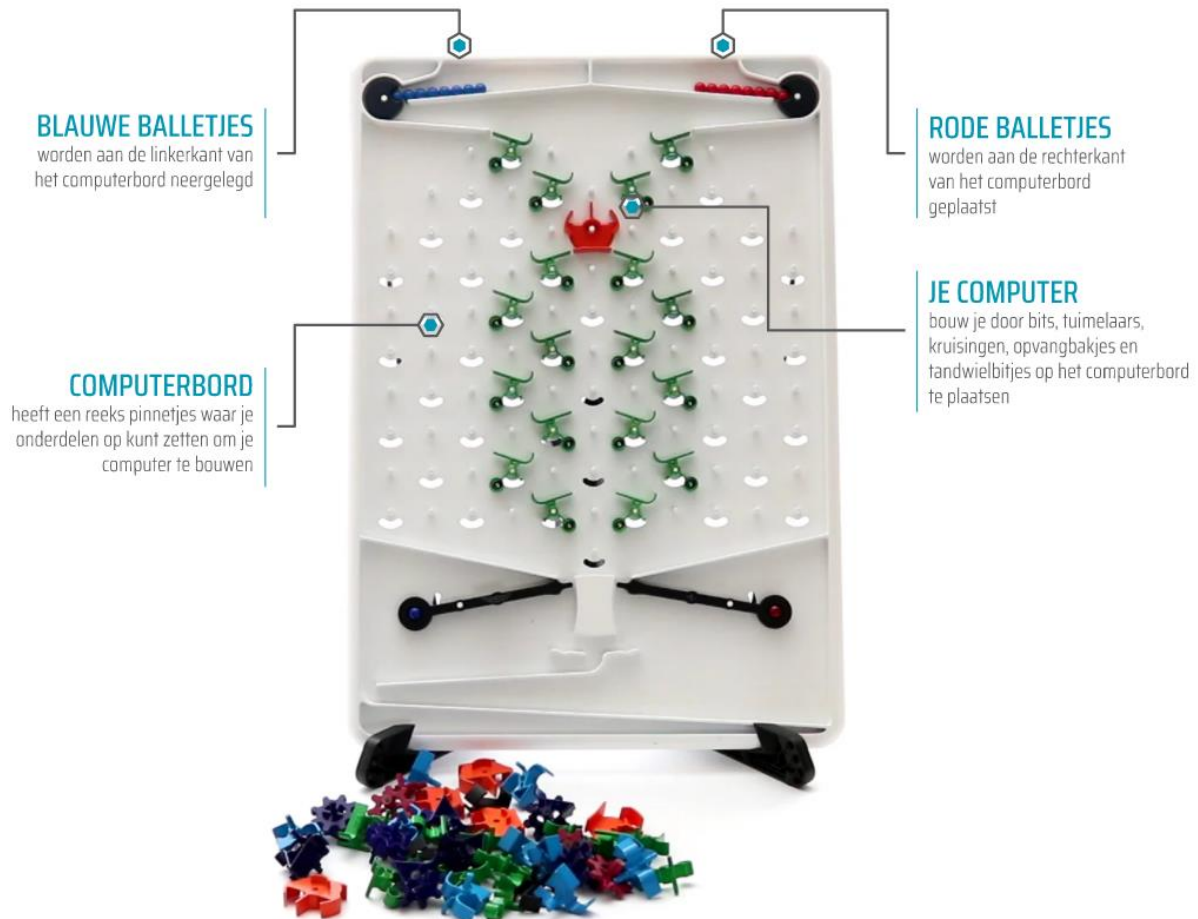
B2. Alle letters

B3. Bomen

B4. Pluszle

# Opgaven C1 en C2. Turing Tumble

Turing Tumble is een revolutionair nieuw spel, waarbij spelers mechanische knikkercomputers bouwen om logische puzzels op te lossen. Het is verslavend leuk en al spelend ontdek je hoe computers werken. Zie voor meer uitleg over het systeem [www.turingtumble.com](http://www.turingtumble.com)



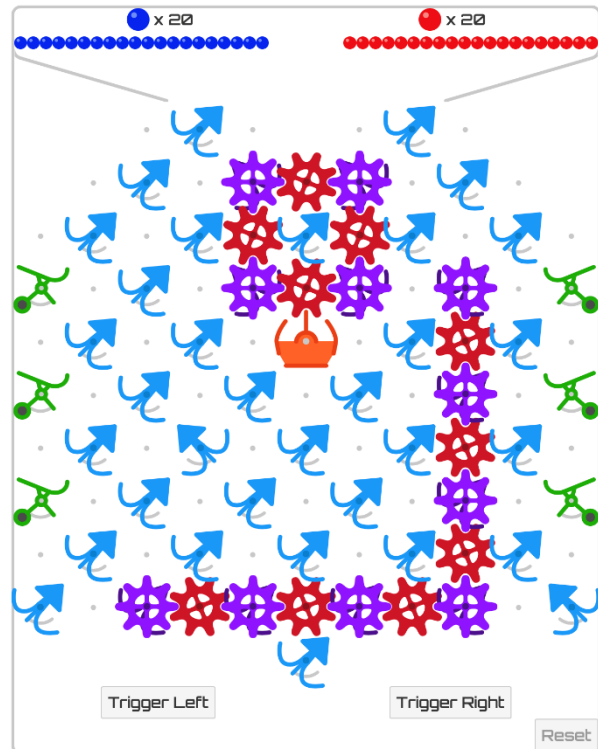
De componenten op bord wordt in deze opgaven weergegeven op de volgende manier:

- > groene tuimelaar die de knikker naar rechts laat vallen
- < groene tuimelaar die de knikker naar links laat vallen
- L blauw bit die de knikker naar links laat vallen
- R blauw bit die de knikker naar rechts laat vallen
- \ paars tandwiel die de knikker naar rechts laat vallen
- / paars tandwiel die de knikker naar links laat vallen
- X oranje cross
- S een stop
- , een plek waar niks zit
- . een lege pin op een positie waar een rood tandwiel kan zitten
- o rood tandwiel

De bijbehorende in- of uitvoer is in de vorm van het bord:

```
R 20 20
.L. .L.
.L.\o\L.
.L.LoLoL.L.
>.L.\o\./.<
.L.L.X.LoL.
>.L.L.L./.<
.L.R.L.LoL.
>.L.L.L./.<
.L.L.L.LoL.
L./o/o/o/.R
L
```

Op de eerste regel staat vermeld met welke kleur knikker wordt begonnen en hoeveel blauwe en rode knikers bovenaan liggen. De regels beginnen zonder voorafgaande spaties.



## Opgave C1. Bepaal de uitvoer

Schrijf een programma dat een beginsituatie van een Turing Tumble knikkercomputer inleest van standaard input. Voor het format zie het voorbeeld hierboven.

Je programma schrijft naar standaard output één regel met daarop de knikers die achtereenvolgens uit de knikkercomputer komen vallen, een R voor een rode knikker en een B voor een blauwe.

Voorbeeld uitvoer:

```
RBBBBBBBRRRRRBBBRRBRRRBBBBBBBBRRRRBRRR
```

De knikers vallen in volgorde van links naar rechts, dus als laatste vallen er vier rode knikers. De laatste rode triggerde een blauwe knikker, maar omdat er bovenin geen blauwe knikers meer over zijn blijven er bovenin twee rode knikers liggen.

Voor je programma geldt een tijdlimiet van 1 seconde.

Voor deze opgave geldt dat voor de eerste zeven testgevallen het maximum aantal ballen aan elke kant 20 is. Voor de laatste drie testgevallen is het maximum aan elke kant 1000.

De functionaliteit van jouw programma moet hetzelfde zijn als van de app op <http://tumble-together.herokuapp.com>.

# Opgave C2. Configureer

Schrijf een programma dat de uitvoer van een Turing Tumble knikkercomputer inleest van standard input. Die uitvoer staat op één regel tekst en bestaat uitsluitend uit de letters R en B. De lengte van deze uitvoer is precies 8 tekens.

Je programma schrijft naar standard output in het format uit het voorbeeld hierboven de configuratie van een knikkercomputer die de gegeven uitvoer genereert.

Voor je programma geldt een tijdlimiet van 2 seconden.

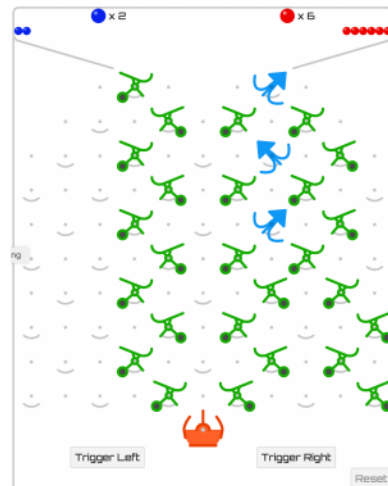
Voorbeeld:

Invoer:

BRRRBRRR

Uitvoer:

```
B 2 6
  .> .L.
  .>.<.>.>
  .>.>.>R.<
  .>.<.>.>.>
  .>.>.>L.<
  .>.<.>.>.>
  .>.>.<.>
  .>.<.>.>.<
  .>.>.<.>
  .>.<.>.<
  X
```



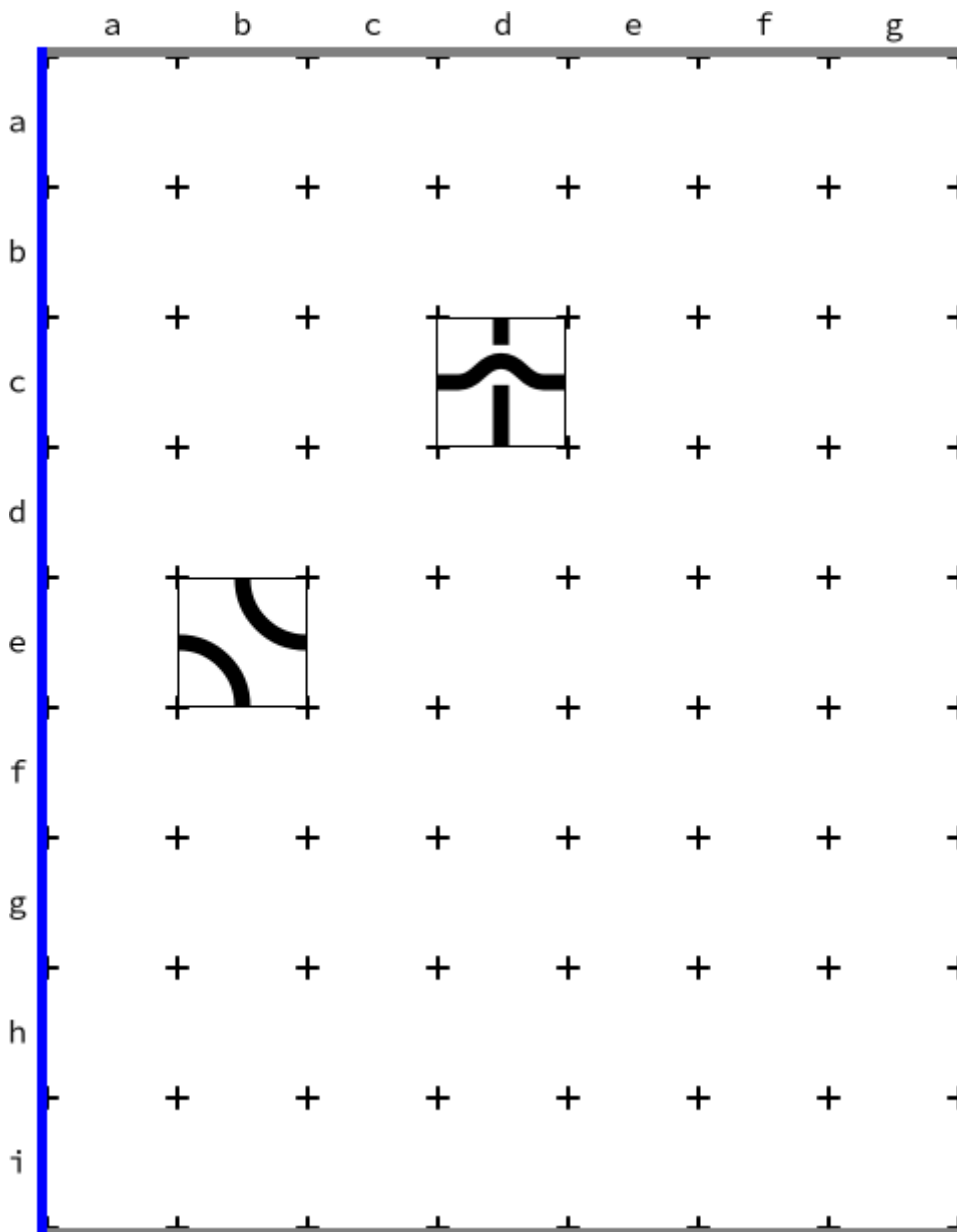
# Opgave D. Spaghetti

## Introductie

Spaghetti is een bordspel voor twee personen. Het is ontwikkeld door Jelmer Firet in samenwerking met het CodeCup team. Spaghetti is gebaseerd op het spel [Black path game](#) dat werd geanalyseerd in het tijdschrift [Pythagoras](#), dat zich richt op leerlingen uit het voortgezet onderwijs.

## Spelbord

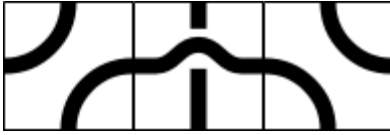
Spaghetti wordt gespeeld op een bord van  $9 \times 7$  vakjes. Aan het begin van het spel zijn deze vakjes leeg, met uitzondering van twee stenen die door de jury zijn gekozen voor die ronde.



*Figuur 1: een mogelijke beginsituatie*



Spelers blauw en rood mogen om beurten zetten; bouw begint. Een zet betekent het plaatsen van een tegel op een vrij vakje. Daarbij is er keuze uit drie soorten tegels:



*Figuur 2: de beschikbare tegels: respectievelijk left ( $l$ ), straight ( $s$ ) en right ( $r$ ).*

Elke tegel bevat twee lijnsegmenten, waarmee twee van de zijden van de tegel worden verbonden. Als het spel vordert worden er paden en cykels (gesloten wegen) gevormd. Deze paden zijn van belang voor de scores.

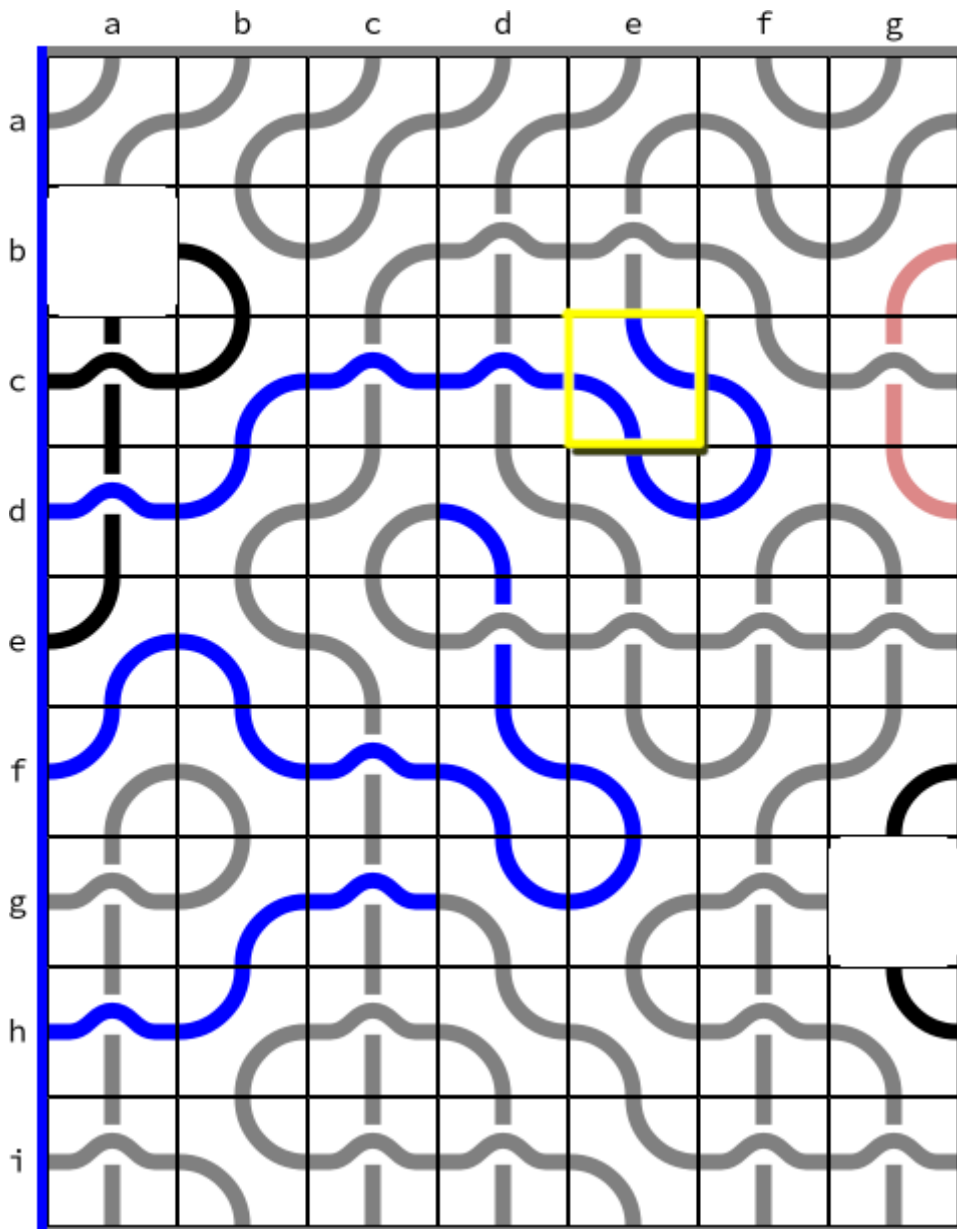
## Punten

Iedere speler begint met 50 punten.

Als een speler een cykel maakt als hij een tegel plaatst verliest die speler 5 punten.

Als een speler een pad vormt van één van de muren naar diezelfde muur verliest hij 3 punten.

Als speler blauw een pad vormt van de linkermuur naar de rechtermuur, krijgt hij voor iedere tegel vanaf de laatste geplaatste tot de linkermuur een punt. Voor speler rood geldt dat van de geplaatste tegel tot de rechtermuur. Als het pad beide segmenten van de tegel bevat tel je het aantal tegels vanaf de muur tot en met het tweede segment. Zie figuur 3 voor een voorbeeld.



*Figuur 3: Als blauw cer spelt (geel aangegeven), ontvangt ze 10 punten. Als rood cer zou hebben gespeeld zou hij 11 punten hebben gekregen. De huidige score is 76 voor blauw en 47 voor rood.*

Omdat een tegel twee segmenten bevat kan het zo zijn dat het plaatsen van een tegel twee paden compleet maakt. Als dat twee verschillende paden zijn wordt de regel voor de puntentelling in zo'n geval tweemaal toegepast.

In de visuele weergave van het spel worden de segmenten van tegels in eerste instantie zwart gekleurd. Als een pad wordt gevormd waarvan het eindpunt aan de boven- of onderkant van het bord ligt, worden de segmenten grijs; ze tellen dan niet mee in de score. Een lichte kleur geeft puntenverlies aan omdat er een cykel is gevormd of een pad dat de eigen kant met zichzelf verbindt. Een blauw of rood segment betekent dat het meetelt in de score voor die speler.

## Invoer en uitvoer

Er is een protocol voor de manier waarop je programma met de jurysoftware moet samenwerken..

Je programma begint met het lezen van twee zetten die de jury heeft gedaan, elke zet op één regel van standard input. Een zet wordt aangegeven met drie kleine letters: de eerste letter geeft de rij aan, de tweede de kolom en de derde geeft aan welk type tegel wordt gespeeld (zie ook Figuur 2). In figuur 1 zijn de twee zetten van de jury `cds` en `ebr`.

Als jouw programma met blauw speelt zal de volgende regel uit het woord "Start" bestaan en daarna moet je programma de eerste zet uitvoer. Vervolgens leest je programma telkens de zet van de tegenstander en voert het de eigen zet uit. Dit gaat zo door tot je programma een regel met het woord "Quit" als invoer leest. Als jouw programma in plaats van het woord "Start" een zet leest, is dat de eerste zet van de tegenstander en speelt je programma met rood. Als je geen zetten meer kunt doen krijgt je programma als laatste invoer het woord "Quit".

Als je programma een ongeldige zet doet of te veel tijd gebruikt krijg je als invoer ook een regel met "Quit". Dat geeft je de mogelijkheid om eventueel nog informatie naar standard error te schrijven om je programma te kunnen debuggen.

Voor meer informatie over in- en uitvoer zie ook de [technical rules](#).

## Voorbeeld van de communicatie in een normaal verlopen spel

Invoer blauw	Uitvoer blauw	Invoer rood	Uitvoer rood
<code>cds</code>		<code>cds</code>	
<code>ebr</code>		<code>ebr</code>	
<code>Start</code>	<code>ids</code>		
		<code>ids</code>	<code>cbl</code>
<code>cbl</code>	<code>egs</code>		
		<code>egs</code>	<code>cas</code>
<code>cas</code>	<code>eds</code>		
<i>etcetera</i>	<i>etcetera</i>	<i>etcetera</i>	<i>etcetera</i>
		<code>cer</code>	<code>ggs</code>
<code>ggs</code>	<code>bal</code>		
<code>Quit</code>		<code>Quit</code>	

## Competitie

In de competitie spelen deelnemers tweemaal tegen elkaar, eenmaal als blauw en eenmaal als rood. De winnaar van het toernooi is de speler met het hoogste puntentotaal.

Als je programma meer tijd gebruikt dan is toegestaan (30 seconden voor het hele spel; de bedenktijd van de tegenstander telt daarbij niet mee), crasht of een ongeldige zet doet, krijg je voor die wedstrijd 0 punten. Het spel wordt dan verder gespeeld door een speler van de jury die random zetten doet, zodat je tegenstander nog altijd punten kan halen. Zie ook de [technical rules](#) voor meer informatie hierover.

## Deelnemen?

Meld je aan op [www.codecup.nl](http://www.codecup.nl) en lees daar alles over deze opgave. Je kunt als je bent ingelogd je programma inzenden en de voorrondes bekijken om te zien hoe je programma zich houdt. In de technische regels staat aangegeven waar je programma aan moet voldoen.

De beste leerling of docent krijgt de Windesheim Digitalisprijs, een geldbedrag van 200 euro.

Als je programma wordt geaccepteerd voor deelname aan het toernooi verdien je 20 punten voor deze opgave. Als je programma zonder fouten speelt kun je daarmee nog eens 50 punten verdienen. De uitslag van de competitie is bepalend voor de laatste 30 punten.