

# De eerste ronde Nederlandse Informatica Olympiade 2017-2018

---



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, begin september 2018 in Tsukuba, Japan.

## 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 2018 wordt gehouden op de Universiteit Twente. Voor deelname aan die 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](http://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.

Je kunt je uitwerkingen uploaden naar [submit.informaticaolympiade.nl](http://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 17 januari worden geupload. Op 20 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](http://www.codecup.nl) en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 20 januari 2018 tegen elkaar uit in een toernooi dat te volgen is op [www.codecup.nl](http://www.codecup.nl) en de beste leerling of docent uit het voorgezet onderwijs wint de jaarlijkse Windesheim Digitalisprijs van 200 euro. Inzenden is tot 20 januari 10.00 u. Dan begint het toernooi.

Voor alle opgaven geldt dat je er van uit mag gaan dat je programma's alleen correcte invoer aangeboden krijgen. Voor de A- en C-opgave wordt je programma getest met het voorbeeld uit de opgave; als dat goed wordt opgelost worden er tien andere testgevallen gebruikt, waarvoor je per testgeval punten kunt krijgen. Als je het voorbeeld niet goed hebt opgelost kun je in het systeem kijken wat er mis is gegaan.

#### 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; zie de link op [www.informaticaolympiade.nl](http://www.informaticaolympiade.nl).

#### 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 het goede antwoord instuurt krijg je 50 punten per opgave. Voor iedere dag later gaat er één punt van je score af. Inzendingen na 17 januari 2018 zullen niet worden verwerkt.

Als je een verkeerd antwoord hebt gegeven, verlies je 10 punten.

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, kan 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 Black Hole kan spelen. Aan dit toernooi doen ook andere deelnemers mee, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 20 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)

# Opgave A1. Rechthoek

---

Schrijf een programma dat twee getallen N en M inleest van standard input. De getallen worden gescheiden door een spatie. Het programma schrijft naar standard output een rechthoek van N regels van elk M keer een \*, zoals te zien is in het voorbeeld hieronder.

Voorbeeld

Invoer: 5 7

Uitvoer: \*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Randvoorwaarde: N en M zijn positieve gehele getallen, met  $0 < N < 24$  en  $0 < M < 80$ .

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave A2. Caesarcode

---

De volgende manier om een bericht te coderen is al bekend van Julius Caesar.



Schrijf de letters van het alfabet op.

Kies een getal  $N$ , bijvoorbeeld 23.

Schrijf onder de letters van het alfabet nog een keer het alfabet op, maar schuif alle tekens  $N$  plaatsen naar links op. Als je voorbij de A komt ga je bij de Z weer verder.

Vervang iedere letter uit het bericht door de letter die eronder staat.

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (boodschap)**  
**X Y Z A B C D E F G H I J K L M N O P Q R S T U V W (code)**

Het woord **COMPUTER** wordt dan in de code met  $N=23$  **ZLJMRQBO**.

Schrijf een programma dat een woord inleest van standard input. Het woord wordt geschreven met hoofdletters. Bijzondere tekens als accenten, spaties en dergelijke worden niet gebruikt. Daarna leest het programma een geheel getal  $N$  in, met  $0 < N < 26$ .

Het programma schrijft één regel uitvoer naar standard output. Op die regel staat het woord uit de invoer, maar als code, waarbij op de manier zoals hierboven uitgelegd alle letters  $N$  posities worden opgeschoven.

Voorbeeld:

Invoer:            COMPUTER  
                      23

Uitvoer:           ZLJMRQBO

Randvoorwaarde:    Het woord in de invoer bestaat uit maximaal 60 karakters.

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave A3. Mieren

Mieren zijn bijzondere beestjes. Als ze bewegen, vormen ze rijen zodat iedere mier zijn voorganger volgt. Het is niet zo bekend wat er gebeurt als twee rijen mieren elkaar treffen die in tegengestelde richting lopen op een randje waar ze elkaar niet kunnen passeren. Er is een theorie die zegt dat in zo'n geval mieren over elkaar heen kunnen springen. Als de rijen elkaar treffen dan gebeurt er elke seconde wat bijzonder is: als twee mieren elkaar aan het begin van de seconde aankijken (dus zonder dat er een andere mier tussen staat) dan wisselen die twee mieren aan het einde van de seconde van plaats.



Schrijf een programma dat van standard input twee positieve gehele getallen N en M inleest. De getallen worden gescheiden door een spatie. Vervolgens wordt een regel met N hoofdletters ingelezen, daarna een regel met M hoofdletters. Iedere hoofdletter staat voor één van de mieren (er zijn er in totaal dus nooit meer dan 26); de eerste regel bevat de mieren die vanaf links komen aanlopen, de tweede regel de mieren die vanaf rechts komen aanlopen. Je leest de mieren in de volgorde waarin ze lopen; de eerst genoemde mier loopt dus voorop! Tenslotte wordt een getal T ingelezen dat aangeeft na hoeveel seconden je de volgorde waarin de mieren nu staan moet geven.

Voorbeeld

Invoer: 3 4  
JLA  
CRUO  
3

Uitvoer: CARLUJO

Toelichting:

Aan het begin staan de mieren zo tegenover elkaar; rode mieren lopen naar rechts, blauwe naar links.

Begin: **ALJCRUO**

Als links een rode mier staat en rechts een blauwe, dan wisselen ze van positie.

Na 1 seconde: **ALCJRUO**

Na 2 seconden: **ACLRJUO**

Na 3 seconden: **CARLUJO**

Voor je programma geldt een tijdslimiet van 2 seconden.

Randvoorwaarde:  $0 \leq T \leq 50$ .

# Opgave A4. Bitcoin

---



De bitcoin is een digitale munteenheid waarvan de waarde sterk op en neer kan gaan.

Van een kennis met voorspellende gaven krijg je de waarde van de bitcoin in euro's voor de komende  $N$  dagen.

Je mag nu proberen zoveel mogelijk winst te maken in die periode. Daarbij geldt het volgende:

- Aan het begin heb je geen bitcoin. Aan het eind van de periode heb je ook geen bitcoin.
- Je mag nooit meer dan één bitcoin tegelijk in je bezit hebben.
- Iedere dag van de periode kun je een bitcoin kopen als je er geen in je bezit hebt, of de bitcoin die je in bezit hebt verkopen.

Doel is aan het einde van de periode zoveel mogelijk winst te hebben gemaakt.

Je programma leest van standard input eerst het getal  $N$ . Daarna leest je programma  $N$  regels, elk met een getal  $W$ , dat de waarde van de bitcoin op de betreffende dag aangeeft.

Je programma schrijft naar standard output één regel met daarop de maximale winst die je kunt halen in deze  $N$  dagen. Dat kan eventueel 0 zijn; je hoeft geen aankoop te doen.

Voorbeeld

Invoer:

```
10
5
11
4
2
8
10
7
4
3
6
```

Uitvoer:

```
17
```

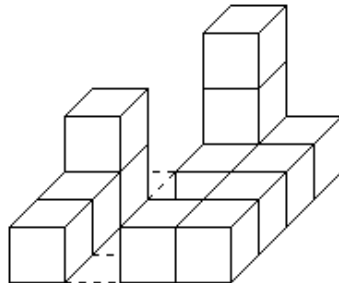
Voor je programma geldt een tijdslimiet van 2 seconden.

Randvoorwaarden: In acht van de tien testgevallen geldt dat  $1 < N < 100$ . Bij één testgeval geldt  $N=2000$  en bij één testgeval geldt  $N=100000$ .

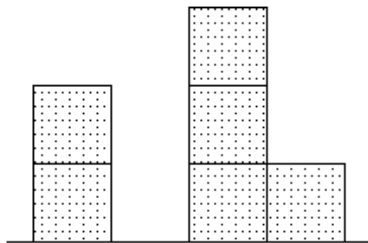
Voor ieder getal  $W$  geldt  $0 < W < 100$ .

# Opgave A5. Blokken

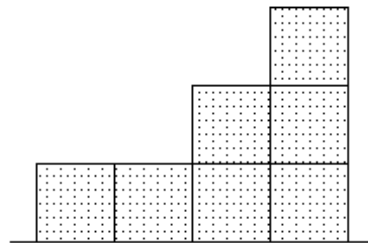
Op een vierkante tafel worden een aantal kubussen gestapeld, netjes in het gelid. Hier zie je een voorbeeld op een tafel van 4 bij 4 blokken:



Je kunt aantekeningen maken van zo'n opstelling, door het vooraanzicht en een zijaanzicht te tekenen. Bij de stapeling van het voorbeeld ziet het er als volgt uit:

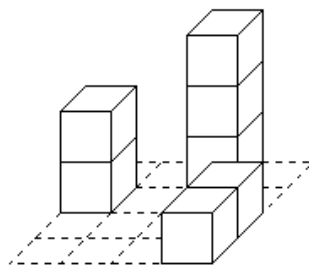


Vooraanzicht

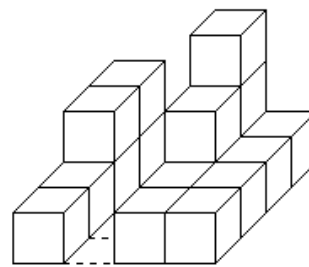


Zijaanzicht vanaf de rechterkant

Je kunt je afvragen of je vanaf deze tekeningen de opstelling kunt reconstrueren. Maar het blijkt dat er dan nog verschillende opstellingen mogelijk zijn. Wel is er bij ieder paar van vooraanzicht en zijaanzicht een minimale opstelling van  $N$  blokken die past bij deze plaatjes, en een maximale opstelling van  $N+M$  blokken.



Minimale opstelling,  $N=7$



Maximale opstelling,  $M=10$



Schrijf een programma dat een getal  $B$  inleest van standard input, met  $0 < B \leq 8$ .  $B$  geeft de afmeting van de vierkante tafel aan, gemeten in blokken.

Daarna worden twee regels ingelezen van elk  $B$  getallen  $H$ , met  $0 \leq H \leq 8$ . De getallen worden gescheiden door spaties.

De eerste regel geeft het aantal blokken boven elkaar in het vooraanzicht.

De tweede regel geeft het aantal blokken boven elkaar in het rechter zijaanzicht.

De invoer is altijd zo dat je er een oplossing bij kunt vinden; je krijgt geen onmogelijke gegevens.

Je programma voert een regel uit met daarop twee getallen  $N$  en  $M$ , gescheiden door een spatie.

$N$  is het minimale aantal blokken dat nodig is voor een opstelling die past bij de invoer.

$M$  is het maximum aantal blokken dat je kunt toevoegen.

Voorbeeld:

Invoer:

```
4
2 0 3 1
1 1 2 3
```

Uitvoer:

```
7 10
```

# Opgave B1 tot en met B4:

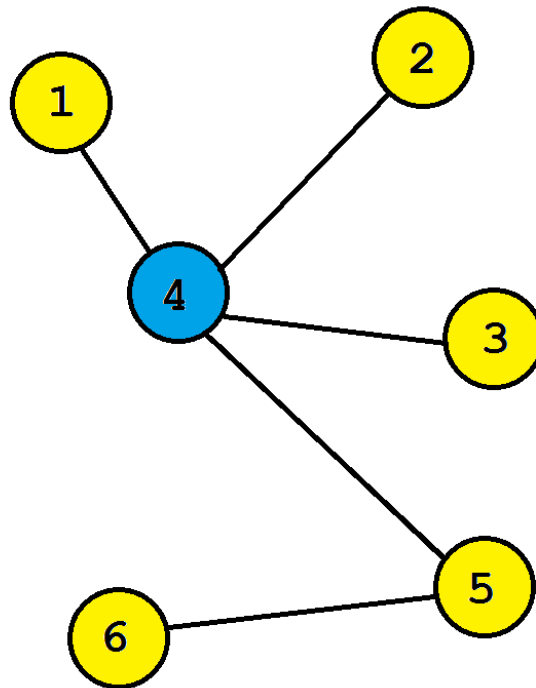
---

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

- B1. Weg van A
- B2. Rondweg
- B3. Staaf
- B4. Rechthoeken

# Opgave C1. Centroïde

---



In de informatica (en de wiskunde) is een boom een structuur die bestaat uit punten en lijnen. Lijnen verbinden punten met elkaar, maar er zijn geen lussen in een boom. Je kunt dus maar op één manier van elk punt naar een ander punt. In de figuur hierboven zie je een boom met zes punten en vijf lijnen.

Aan ieder punt van een boom hangen één of meer lijnen. Omdat we ze vanuit een punt benaderen noemen we dat ook wel de takken. Aan punt 5 zitten bijvoorbeeld twee takken.

Het gewicht van een tak ten opzichte van een punt is het aantal lijnen dat er aan die kant van het punt in de boom zitten. Het gewicht van de tak van punt 5 richting punt 6 is 1, omdat daar alleen de lijn van 5 naar 6 zit; het gewicht van de tak van punt 5 richting punt 4 is 4, omdat aan die kant van punt 5 zich 4 lijnen van de boom bevinden.

Voor elk punt geldt dat de som van de gewichten van de takken gelijk is aan het aantal lijnen in de boom. Bij het voorbeeld is die som 5.

De centroïde van een boom is het punt of zijn de twee punten waarvan het gewicht van de zwaarste tak zo klein mogelijk is. (Je kunt bewijzen dat er niet meer dan twee punten zijn met zo'n minimaal hoogste gewicht.)

In het voorbeeld is dat punt 4; de zwaarste tak vanuit 4 heeft een gewicht van 2. De zwaarste tak vanuit punt 5 heeft een gewicht van 4; voor de punten die met precies een tak verbonden zijn geldt dat het gewicht van die tak vanuit dat punt steeds 5 is."

Schrijf een programma dat de gegevens over een boom inleest van standard input.

Op de eerste regel staat een getal  $N$ , dat aangeeft dat de boom bestaat uit de punten 1 tot en met  $N$ . Op de volgende  $N-1$  regels staan telkens twee getallen, gescheiden door een spatie. Hiermee worden de  $N-1$  lijnen aangegeven. De volgorde van die twee getallen doet er niet toe.

Je programma moet de centroïde van de boom uit de invoer bepalen. De centroïde bestaat altijd uit één of twee punten. Je programma voert dat punt of die punten (in oplopende volgorde, gescheiden door een spatie) uit naar standard output.

Voorbeeld:

```
Invoer      6
             1 4
             2 4
             4 3
             5 4
             6 5
```

```
Uitvoer     4
```

Randvoorwaarde:  $0 < N \leq 150$ .

Voor je programma geldt een tijdslimiet van 2 seconden.

# Opgave C2. Puzzel

De Informatica Olympiade organisatie is in het bezit van een puzzel. Deze bestaat uit een bord met ergens een pin erin en negen verschillende puzzelstukjes.



De vorm en de nummering van deze negen puzzelstukjes wordt weergegeven in de oplossing van het voorbeeld. Stukje 1 bestaat dus uit een blokje van 1 bij 5 vierkantjes. De hoogte van de puzzelstukjes is niet van belang.

Het is altijd mogelijk om de pin op één van de 49 beschikbare posities te plaatsen en dan de puzzel op te lossen. Je mag de stukjes wel draaien, maar niet spiegelen.

De posities zijn als volgt genummerd.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Schrijf een programma dat van standard input de positie inleest van de geplaatste pin. Schrijf naar standard output een mogelijke oplossing van de bijbehorende puzzel. De positie van de pin wordt met 0 aangegeven.

Voorbeeld

Invoer: 3

Uitvoer: 9904444  
5566643  
5556633  
5522633  
8222773  
8822277  
8111117

Er geldt voor deze opgave een tijdslimiet van 5 seconden.

9	9	0	4	4	4	4
5	5	6	6	6	4	3
5	5	5	6	6	3	3
5	5	2	2	6	3	3
8	2	2	2	7	7	3
8	8	2	2	2	7	7
8	1	1	1	1	1	7

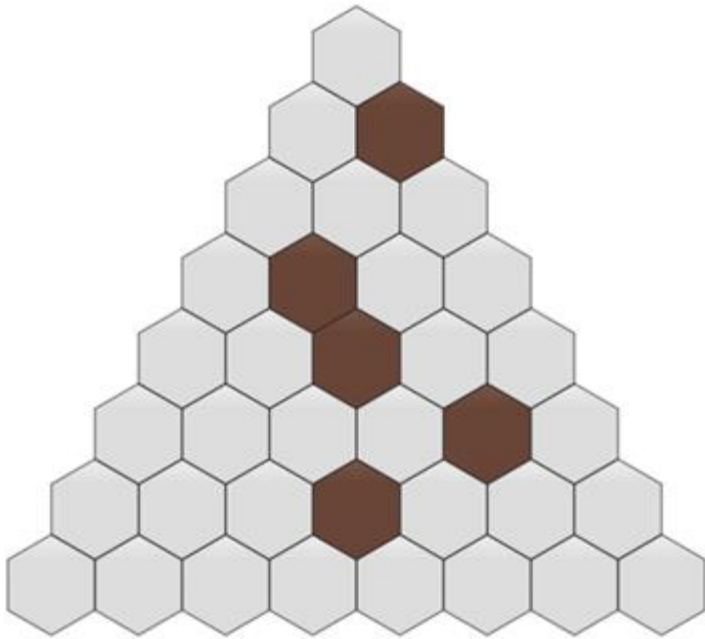
# Opgave D. Black Hole

---



## Inleiding

Black Hole wordt gespeeld op 36 zeshoekige tegels die zijn geplaatst in de vorm van een driehoek. Deze tegels zijn leeg aan het begin van het spel, op vijf bruine tegels na. Als het spel begint kiest het juryprogramma welke vijf tegels bruin worden; tijdens een speelronde in het toernooi bij deze opgave is dat voor iedere wedstrijd dezelfde indeling.



### Spelverloop

De speler met de rode stenen moet beginnen. Beide spelers zetten om beurten, en het spel is afgelopen na in totaal 30 zetten. Aan het begin van het spel hebben beide spelers 15 stenen van hun kleur, rood of blauw, elk met de getallen 1 tot en met 15 erop. In iedere zet plaatst een speler één van de stenen die nog niet gespeeld zijn op een vrije tegel; uiteraard zijn de bruine tegels niet beschikbaar.

Als gevolg van deze regels is er aan het einde van het spel één lege tegel over. Er zijn 36 tegels, 5 zijn er bruin, 15 rood en 15 blauw. De tegel die over is heet het zwarte gat of het Black Hole. En omdat dat een zwart gat is, worden alle aangrenzende tegels dit gat ingezogen.

### Score

De uitslag van het spel hangt af van het aantal punten dat in het zwarte gat terecht komt. Als R het aantal punten is op de rode stenen op de tegels die grenzen aan het zwarte gat, en B het aantal punten op de blauwe stenen.

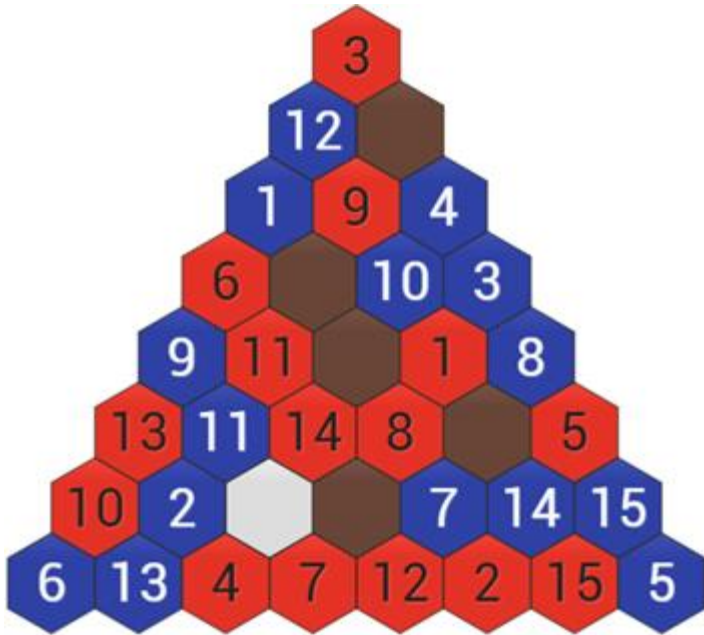
De rode speler krijgt dan  $75 + R - B$  punten en de blauwe speler  $75 - R + B$ .

In het toernooi speel je eenmaal met rood en eenmaal met blauw tegen een tegenstander. Het toernooi wordt gewonnen door de speler met de meeste punten.

Voorbeeld

Kijk eens naar de volgende eindpositie:





Rood heeft drie stenen om het zwarte gat (dat in de afbeelding wit is). Deze stenen hebben de waarden 4, 7 en 14, dus  $R = 4 + 7 + 14 = 25$ .

Blauw heeft twee stenen met de waarden 2 en 11 om het zwarte gat. Dus  $B = 2 + 11 = 13$ .

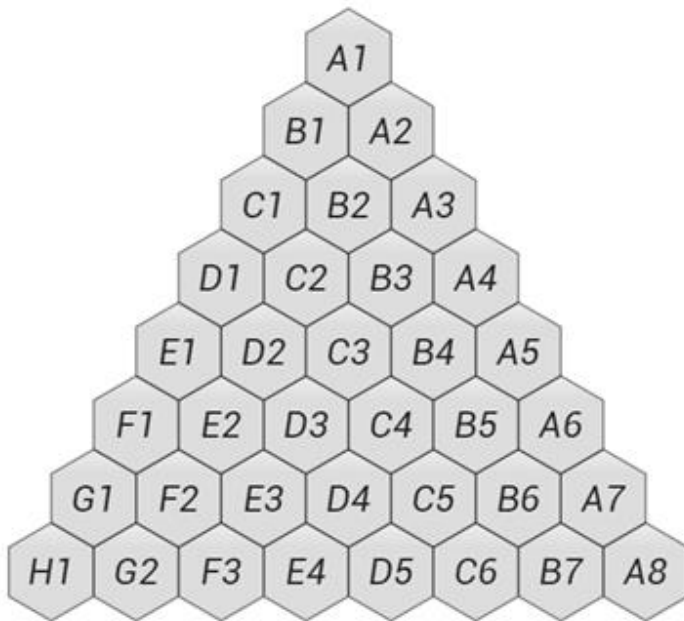
Dus rood verdient deze wedstrijd  $75 + 25 - 13 = 87$  punten en blauw  $75 + 13 - 25 = 63$  punten.

Als je programma de tijdslimiet overschrijdt of een onreglementaire zet doet, zoals het plaatsen van een steen op een tegel die al bezet is of niet bestaat, dan krijg je voor die wedstrijd 0 punten. De tegenstander maakt het spel dan af tegen een programma van de jury. Zie ook de technische regels voor meer informatie.

### In- en uitvoer

Er is een protocol om ervoor te zorgen dat je programma met de jurysoftware kan communiceren. Je leest de zetten van de tegenstander van standard input. En je schrijft de eigen zetten naar standard output. Je mag gebruik maken van standard error.

De tegels hebben elk een naam; zie hiervoor de afbeelding hieronder.



Aan het begin van het spel lezen beide spelers vijf regels als invoer. Iedere regel geeft de positie aan van één van de bruine tegels (bijvoorbeeld "B4").

De rode speler krijgt daarna een regel met het woord "Start" als invoer. Vervolgens is voor beide spelers het ritme: zet bedenken en uitvoeren, volgende zet van de tegenstander inlezen, totdat je van de jury het woord "Quit" te lezen krijgt. De blauwe speler krijgt na de regels met de vijf bruine tegels meteen de zet van de tegenstander als invoer. De jury geeft je het woord "Quit" als het spel is afgelopen of als je een ongeldige zet hebt gedaan.

Een zet heeft het volgende format: <positie>=<getal>. De zet "A3=6" plaatst de steen met waarde 6 op tegel A3.

Voorbeeld van communicatie:

Invoer Rood	Uitvoer Rood	Invoer Blauw	Uitvoer Blauw	Toelichting
A2		A2		Bruine tegel
C2		C2		Bruine tegel
C3		C3		Bruine tegel
B5		B5		Bruine tegel
D4		D4		Bruine tegel
Start				Rood begint
	A1=3	A1=3		Zet van rood
C1=1			C1=1	Zet van blauw
	B7=15	B7=15		Zet van rood

Je programma heeft 5 seconden de tijd om een spel te spelen. De tijd die de tegenstander heeft wordt niet meegeteld.







## 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. Je staat dan in de deelnemerslijst met de vermelding NIO. Zie je nog CodeCup achter je naam, geef dan even een seintje aan [info@informaticaolympiade.nl](mailto:info@informaticaolympiade.nl) om te melden dat je als leerling of docent deelneemt.

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.

## Sponsors Nederlandse Informatica Olympiade:

 Eljakim IT BV	 Rijksuniversiteit Groningen	 Vrije Universiteit Amsterdam	 Technische Universiteit Eindhoven
 Universiteit Leiden	 ASML	 Universiteit Twente	 Christelijke Hogeschool Windesheim
 SLO	 COMMIT/	 Universiteit Utrecht	 Acer
 Ministerie van Onderwijs, Cultuur en Wetenschap	 Technische Universiteit Delft	 Radboud Universiteit Nijmegen	