

De eerste ronde Nederlandse Informatica Olympiade 2019-2020



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 2020 in Singapore.

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 2020 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

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. Voor nadere details verwijzen we je graag naar onze [privacyverklaring](#).

Je kunt je uitwerkingen uploaden naar 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 15 januari worden geüpload. Op 18 januari wordt de eerste ronde gejureerd en kort daarna worden de uitslagen gepubliceerd.

Voor de spelopgave, opgave D, moet je je aanmelden op onze website www.codecup.nl en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 18 januari 2020 tegen elkaar uit in een toernooi dat te volgen is op www.codecup.nl. De beste leerling of docent uit het voorgezet onderwijs wint de jaarlijkse Windesheim Digitalisprijs van 200 euro. Inzenden mag tot 18 januari 7.00 u. Dan begint de wedstrijdcompetitie.

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. 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 15 januari 2020 worden niet 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 Gomoku kan spelen. Aan dit toernooi doen ook andere deelnemers me, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 18 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

Opgave A1. Hoogste uitkomst

Schrijf een programma dat drie getallen N, M en V inleest, elk van een regel van standard input. Het programma schrijft naar standard output het grootste van de volgende getallen:

$$N * (M + V)$$

$$M * (N + V)$$

$$V * (N + M)$$

Voorbeeld

Invoer: 5
 7
 2

Uitvoer: 49

Toelichting:

De drie getallen zijn:

- $5 * (7 + 2) = 45$
- $7 * (5 + 2) = 49$
- $2 * (5 + 7) = 12$

Randvoorwaarde:

N, M, en V zijn positieve gehele getallen, allemaal kleiner dan 100.

Voor je programma geldt een tijdslimiet van 2 seconden.

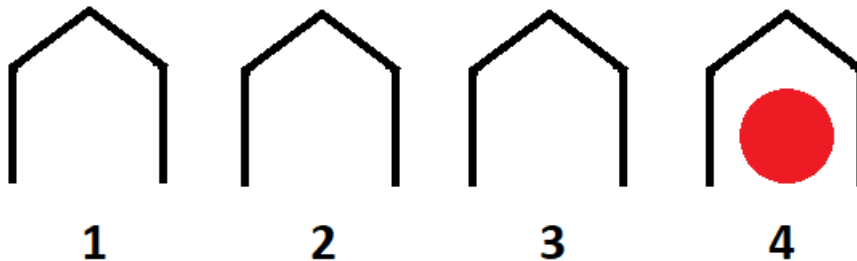
Opgave A2. Balletje-balletje

Misschien heb je het wel eens in actie gezien. Een goochelaar heeft een rij bekertjes en onder één ervan legt hij een balletje. Daarna verwisselt hij razendsnel telkens twee bekertjes. Aan jou de vraag onder welke beker het balletje nu ligt.

Meestal ben je niet in staat om het zo snel te volgen. Gelukkig heb je een hoge snelheidscamera waardoor je een precieze lijst van verwisselingen kunt opstellen.

De goochelaar heeft N bekertjes, die hij neerzet op de posities 1 tot en met N . Er zijn minimaal 4 bekertjes en maximaal 100 bekertjes.

Het balletje legt hij in eerste instantie onder het bekertje op positie N .



Jouw programma bepaalt op basis van de verwisselingen waar het balletje uiteindelijk ligt.

Schrijf een programma dat van standard input een regel inleest met het getal N . Vervolgens komt er een getal W dat aangeeft hoeveel verwisselingen er zullen zijn. Er is minimaal 1 verwisseling; er zijn maximaal 1.000.000 verwisselingen. Daarna leest je programma W regels met op iedere regel één verwisseling, aangegeven met twee verschillende posities, gescheiden door een spatie.

Je programma schrijft naar standaard output één regel, met daarop de positie waarop het balletje ligt na alle verwisselingen.

Voorbeeld

```
Invoer: 4
         4
         1 2
         3 4
         4 1
         2 3
```

```
Uitvoer: 2
```

Voor je programma geldt een tijdslimiet van 5 seconden.

Opgave A3. N-draai

Een simpele manier om een zin er iets geheimzinniger uit te laten zien is gebruik te maken van een procedé dat bekend staat als de N-draai. Voor ieder woord in de zin kijk je of het met een N begint of eindigt. Als dat zo is, draai je het woord om. Alle andere woorden en alle spaties laat je onveranderd.

Als je een zin te lezen krijgt waarop een N-draai is toegepast kun je makkelijk de oorspronkelijke zin terugvinden; je past nog een keer de N-draai toe.

Schrijf een programma dat een zin inleest van standard input. De zin wordt geschreven met hoofdletters. Spaties scheiden de woorden; accenten en andere tekens worden niet gebruikt.

Het programma schrijft dezelfde zin na het uitvoeren van één N-draai op één regel naar standard output.

Voorbeeld:

Invoer: MET COMPUTERS IS HET NOOIT SAAI WERKEN

Uitvoer: MET COMPUTERS IS HET TIOON SAAI NEKREW

Randvoorwaarde: In de invoer staat een zin van maximaal 80 tekens.

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A4. Objectherkenning

Bewakingscamera's worden steeds vaker gebruikt. Maar omdat het meestal te duur is om er iemand de hele dag naar te laten kijken worden de opnames bewaard om ze terug te kunnen zien als er iets bijzonders blijkt te zijn gebeurd.



Daarom zou het handig zijn als je een computer kon gebruiken om automatisch te kunnen ontdekken als er iets in beeld gebeurt. Een eenvoudige techniek hiervoor heet achtergrondverwijdering. De computer kent het achtergrondbeeld, een plaatje van de situatie als er niets gebeurt. Als de computer een nieuw beeld krijgt van de bewakingscamera, vergelijkt hij elke pixel in het beeld met de overeenkomstige pixel in het achtergrondbeeld. Als de kleur er erg op lijkt wordt de pixel gemarkeerd als achtergrond met een 0, anders als voorgrond met een 1. Dit noemen we het voorgrondbeeld. Grotere gebieden van voorgrondpixels geven een aanwijzing dat er iets gebeurt (bijvoorbeeld een persoon die in het zicht van de camera komt of een voorwerp dat is achtergelaten).

Het belang B van een rechthoekig gebied in een voorgrondbeeld kan worden berekend met de volgende formule:

$$B = \frac{v * v}{O}$$

waar 'v' het aantal voorgrondpixels is in het gebied en O de oppervlakte van het gebied.

Schrijf een programma dat een voorgrondbeeld inleest van standard input. Op de eerste regel staan twee getallen R en K, het aantal rijen en kolommen van het voorgrondbeeld, gescheiden door een spatie. R en K zijn allebei minstens 4 en hoogstens 50. Daarna volgen er R regels met elk K tekens, uitsluitend 0 en 1 worden hierbij gebruikt.

Jouw programma moet het rechthoekige gebied vinden waarvoor het belang zo groot mogelijk is. Je programma schrijft naar standard output één regel met daarop vier getallen X, Y, S en T, gescheiden door spaties.

X is de kolom van de linkerbovenhoek van de rechthoek

Y is de rij van de linkerbovenhoek van de rechthoek

S is het aantal kolommen van de rechthoek

T is het aantal rijen van de rechthoek

Je moet een rechthoek uitvoeren waarvan het belang maximaal is. Als er meerdere rechthoeken zijn met een gelijk belang hoef je er maar één uit te voeren.

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

Voorbeeld:

Invoer: 8 9
 00000000
 01000000
 00000000
 00011000
 00001100
 00000000
 00000000
 01111000
 00001000

Uitvoer: 2 8 4 1

Toelichting: De rechthoek met als bovenhoek kolom 2 en rij 8 met als breedte 4 kolommen en hoogte 1 rij heeft een belang van 4. Immers $v=4$, $O=4$ en het belang is $4*4/4 = 4$.

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A5. Wildcards

Een wildcard kun je gebruiken om een patroon in een tekenreeks aan te geven. Meestal worden de volgende wildcards gebruikt:

- ? kan worden vervangen door een willekeurig teken
- * kan worden vervangen door een willekeurige tekenreeks (zelfs van lengte 0)

In deze opgave krijg je een patroon met wildcards te zien en daarna een serie tekenreeksen; jouw programma moet bepalen of elk van deze reeksen past bij het aangegeven patroon.

Een tekenreeks in de invoer is een string van minstens 1 en hoogstens 20 tekens, waarin de symbolen ? en * niet voorkomen.

Een patroon is een string van minstens 1 en hoogstens 20 tekens waarin de symbolen ? en * wel voor kunnen komen. Elk teken en symbool mag eventueel meerdere keren voorkomen.

Schrijf een programma dat zes regels inleest van standard input.

Op de eerste regel staat een patroon. Daarna volgen vijf regels met telkens één tekenreeks.

Je programma moet voor elk van de tekenreeksen bepalen of deze past bij het aangegeven patroon.

Je programma voert vijf regels uit naar standard output, met op ieder regel het woord “ja” of het woord “nee”.

Voorbeeld:

Invoer	ab*d? abc abcdtje abdy absurde abacusrekenkunde
--------	--

Uitvoer	nee nee ja ja ja
---------	------------------------------

Voor je programma geldt een tijdslimiet van 5 seconden.

Alleen als alle vijf tekenreeksen goed worden beoordeeld door je programma krijg je 4 punten voor de test. Er zijn 10 testgevallen.

Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Op en neer

B2. Boggle

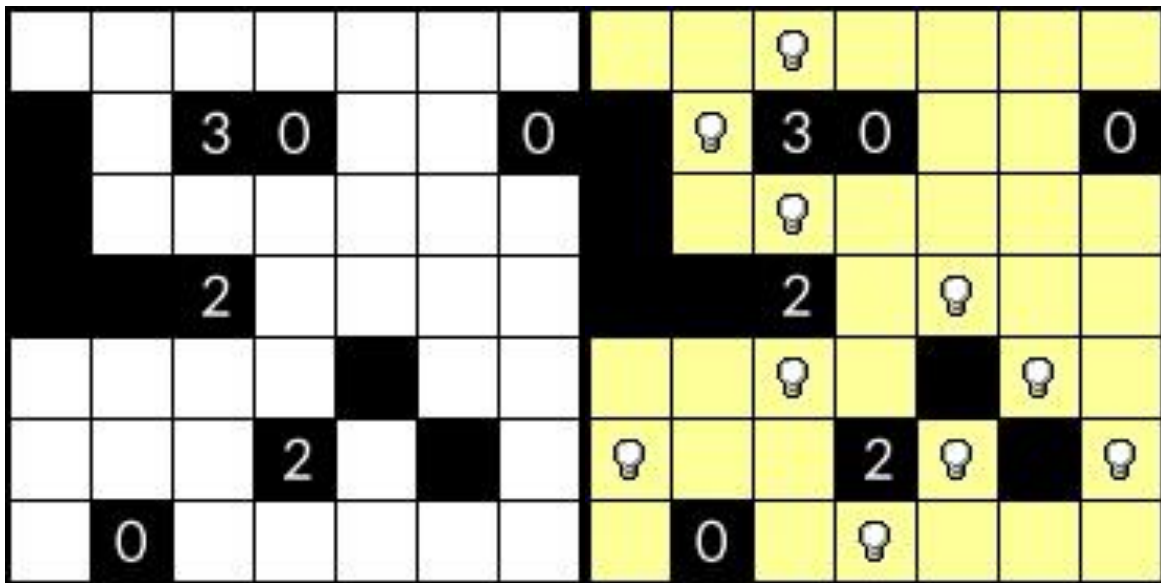
B3. Cocktailsort

B4. Schaakstukplaatsing

Opgave C1. Peertjes

Peertjes is een puzzel van Nikoli. Het wordt gespeeld op een rechthoekig raster van witte en zwarte vakjes. De speler plaatst lichten in de witte vakjes volgens de onderstaande regels:

- Plaats "lichtjes" in witte vakjes zodat al de witte vakjes in het hele raster verlicht worden.
- Twee "lichtjes" mogen NOOIT op elkaar schijnen.
- Een "lichtje" zendt lichtstralen horizontaal en verticaal, tenzij het wordt geblokkeerd door een zwart vakje.
- Een zwart vakje bevat mogelijk een getal van 0 tot 4 dat aangeeft hoeveel "lichtjes" naast de vier zijden moeten geplaatst worden.



Voor deze opgave schrijf je een programma dat een Peertjes-puzzel inleest. Jouw programma voert de oplossing van de puzzel uit naar standaard output.

Je programma leest van standaard input eerst een regel met twee getallen N en M, gescheiden door een spatie. N en M zijn gehele getallen tussen 3 en 13. Deze getallen geven de afmetingen van het raster aan.

Vervolgens leest je programma N regels van elk M tekens. Deze hebben de volgende betekenis:

w	Wit vakje
z	Zwart vakje zonder getal erin
0, 1, 2, 3, 4	Zwart vakje met daarin het aangegeven getal

Je programma schrijft naar standaard output N regels van elk M tekens. Deze hebben de volgende betekenis:

w	Wit vakje
P	Peertje (lichtje)
z	Zwart vakje zonder getal erin
0, 1, 2, 3, 4	Zwart vakje met daarin het aangegeven getal

Voorbeeld:

Invoer: 7 7
wwwwwww
zw30ww0
zwwwwww
zz2wwww
wwwwzww
www2wzw
w0wwwww

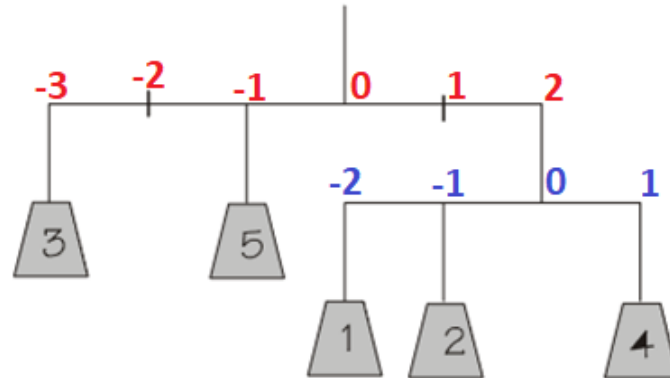
Uitvoer: wwPwwww
zP30ww0
zwPwwww
zz2wPww
wwPwzPw
Pww2PzP
w0wPwww

Je programma moet vijf testgevallen oplossen waarbij N en M niet groter zijn dan 8, en vijf testgevallen waarbij N en M maximaal 14 kunnen zijn. Per goed opgelost testgeval krijg je 10 punten.

In alle testgevallen is er maar één unieke oplossing.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave C2. Mobiel



In de afbeelding zie je een mobiel, precies in evenwicht. Je kunt dat controleren met de volgende berekening:

$$(-3 * 3) + (-1 * 5) + 2 * (1 + 2 + 4) = 0$$

Op dezelfde manier kun je ook van de submobiel controleren dat deze in balans is:

$$(-2 * 1) + (-1 * 2) + (1 * 4) = 0$$

De structuur van een mobiel wordt gegeven met een string. Die van het voorbeeld is:

```
(-3, -1, 2 (-2, -1, 1))
```

Jij moet bij die structuur de gewichten in de mobiel bepalen en die weergeven met een string. Die bij dit voorbeeld is:

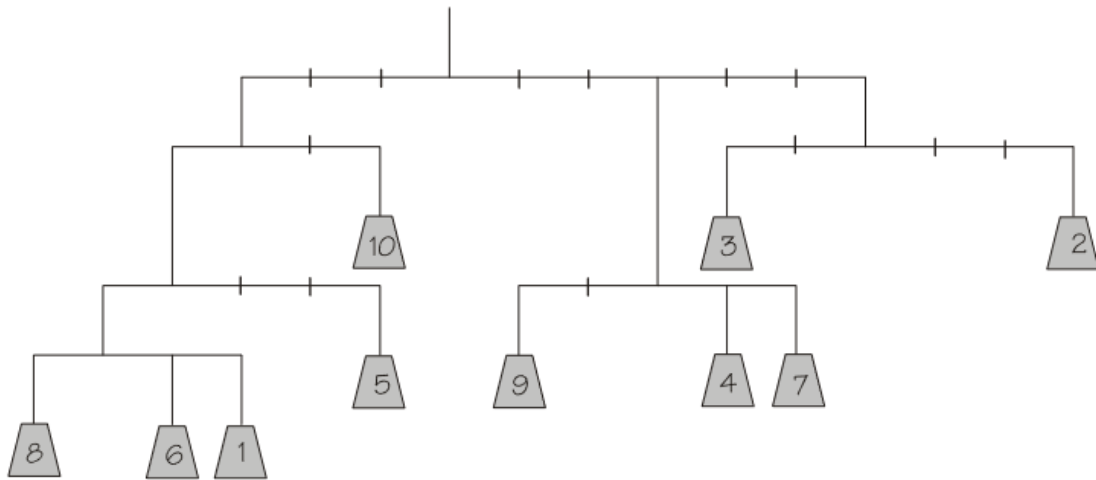
```
(3, 5, (1, 2, 4))
```

Schrijf een programma dat van standaard input één regel inleest, met daarin de structuur van een mobiel.

Jouw programma schrijft naar standaard output één regel met daarop de gewichten in de mobiel in de aangegeven vorm. Er gelden de volgende randvoorwaarden:

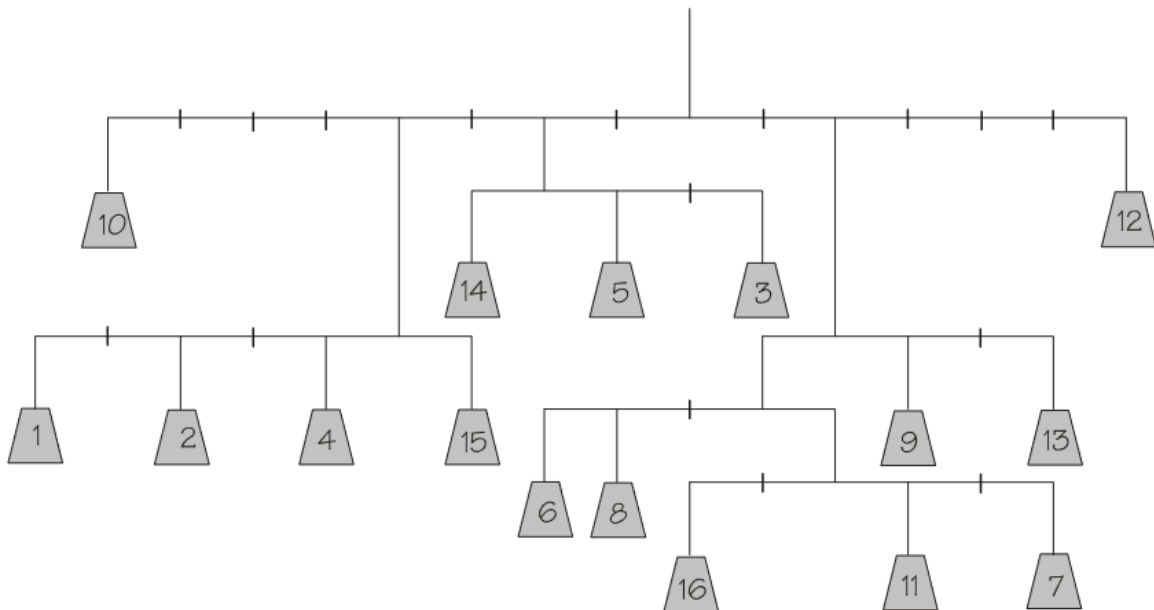
- Alle gewichten zijn een positief, geheel en uniek. Als er n gewichten in de mobiel hangen, hebben die de gewichten van 1 tot en met n.
- Er geldt $n < 17$.
- De getallen in de invoer zijn altijd groter dan -50 en kleiner dan +50.
- In de uitvoer mogen geen spaties worden gebruikt.
- Er is altijd minstens één oplossing bij de gegeven invoer. Als er meerdere goede oplossingen zijn hoeft je er maar één uitvoeren.

Hieronder nog twee voorbeelden met in- en uitvoer en een tekening.



Invoer: $(-3(-1(-1(-1, 1, 2), 3), 2), 3(-2, 1, 2), 6(-2, 3))$

Uitvoer: $((((8, 6, 1), 5), 10), (9, 4, 7), (3, 2))$



Invoer:

$(-8, -4(-5, -3, -1, 1), -2(-1, 1, 3), 2(-1(-3, -2, 1(-2, 1, 3)), 1, 3), 6)$

Uitvoer:

$(10, (1, 2, 4, 15), (14, 5, 3), ((6, 8, (16, 11, 7)), 9, 13), 12)$

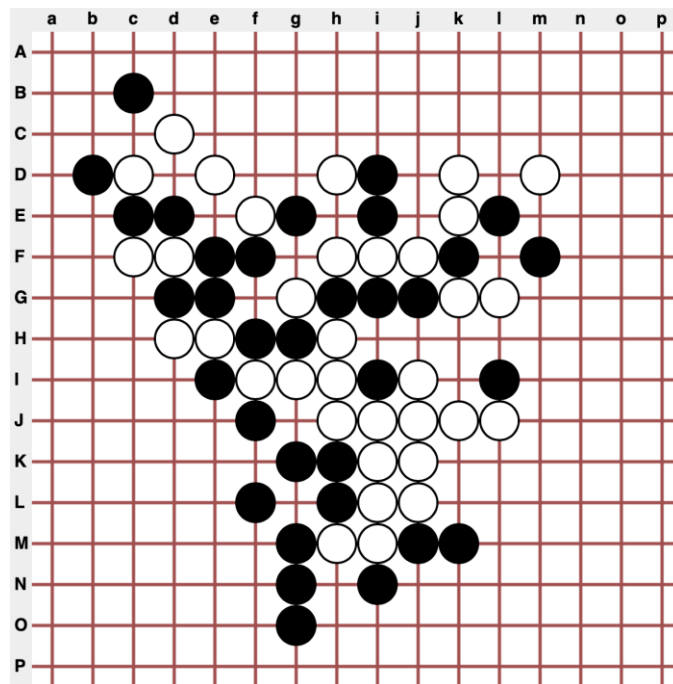
Voor de opgave geldt bij ieder testgeval een tijdlimiet van 5 seconden.

Opgave D. Gomoku

Gomoku, ook wel bekend als Vijf op een Rij, is een strategisch bordspel. Het wordt gespeeld met zwarte en witte stenen op een bord met 16x16 knooppunten. De spelers plaatsen om de beurt een steen van hun eigen kleur op één van de knooppunten op het bord. De speler die het eerst vijf (of meer) aaneengesloten stenen van eigen kleur op een verticale rij, horizontale rij of een diagonaal weet te krijgen wint het spel.

Spelverloop

De zwarte speler begint met het plaatsen van drie stenen: eerst een zwarte steen, dan een witte steen en dan tenslotte weer een zwarte steen. Dan is het aan de witte speler om te beslissen of hij met deze stand op het bord wil doorspelen of niet. Als hij dat wil, gaat het spel gewoon door totdat er vijf stenen van een kleur op een rij op het bord staan. Anders worden de drie eerder geplaatste stenen vervangen door stenen van de andere kleur en moet de zwarte speler de volgende zet doen. Het weigeren van de beginopstelling geldt als een zet, aangeduid als "Zz". Daarna spelen wit en zwart om en om een steen op het bord totdat iemand vijf op een rij heeft.



In de afbeelding zie je een mogelijk eindpositie van een spelletje. De witte speler heeft gewonnen met vijf stenen in rij in rij J.

Score

Wanneer geen van de spelers wint, eindigt het spel in een gelijkspel met als score 50 – 50. De speler die wint krijgt een beloning van $101 - \text{floor}(\text{nummer}/5)$ punten. De verliezer krijgt de rest van $\text{floor}(\text{nummer}/5) - 1$ punten. In deze berekeningen is *nummer* het aantal zetten dat is gespeeld. De floor van een getal is de naar beneden afgeronde waarde van dat getal.

Input/output

Er is een protocol om ervoor te zorgen dat je programma met de jurysoftware kan communiceren. Je leest de zetten van de tegenstander van standaard input. Je schrijft de eigen zetten naar standaard output. Je mag gebruik maken van standaard error; de eerste 10000 tekens per spelletje zijn terug te lezen als je bent ingelogd op de wedstrijdsite.

De zwarte speler krijgt aan het begin van het spel een regel met het woord "Start" als invoer. De zwarte speler speelt vervolgens de eerste drie zetten waarna de witte speler dit, na inlezen, accepteert of weigert met de zet "Zz". Daarna spelen zwart en wit om en om een zet totdat er een winnaar is, of dat het bord vol staat of totdat je van de jury de opdracht "Quit" ingelezen krijgt. De zwarte speler krijgt aan het begin van het spel de eerste zet van de tegenstander ingelezen als invoer. De jury geeft je het woord "Quit" als het spel is afgelopen of als er een ongeldige zet gedaan is. Een zet ziet eruit als "Hj". Bij de zet "Hj" wordt een steen geplaatst in de kleur van de speler die aan de beurt is op de positie Hj. Alleen bij de tweede zet van zwart wordt een witte steen geplaatst op het bord.

Als je programma een illegale zet doet of uit de tijd loopt, dan krijg je de "Quit" instructie toegezonden. Dit geeft je nog de mogelijkheid om debug informatie naar standard error te schrijven.

Voorbeeld van communicatie:

Invoer Zwart	Uitvoer Zwart	Invoer Wit	Uitvoer Wit	Toelichting
Start				Zwart begint dus
	Ek			Zwarte steen
	Fj			Witte steen
	Dj			Zwarte steen
		Ek		
		Fj		
		Dj		
			Zz	Wit weigert dit
Zz				
	F1			Zwart gaat door
		F1		
			Ci	Reactie van Wit
C1				
	Fk			Enzovoort...

Je programma heeft in totaal 5 seconden de tijd om een spel te spelen; dit geldt dus voor alle zetten samen. De tijd die de tegenstander heeft wordt niet meegeteld.

Competitie

Tijdens de competitie speel je tegen iedere tegenstander twee keer, eenmaal met zwart en eenmaal met wit. De winnaar van de wedstrijd is de speler met de meeste punten. Wanneer je programma een ongeldige uitvoer geeft, crasht of uit de tijd loopt krijg je 0 punten. De wedstrijd wordt voortgezet met een speler, uitgevoerd door de jurysoftware, om je tegenstander de gelegenheid te geven om nog punten te kunnen scoren. Die speler is niet bijster slim, die zet willekeurig ergens een steen op het bord en zou dus eventueel per ongeluk kunnen winnen.

Deelnemen?

Meld je aan op 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. Zodra wij merken dat jij een geldige inzending hebt gedaan, zullen we je status van de wedstrijd veranderen van 'CodeCup' in 'NIO', waarmee aangegeven wordt dat jij zowel met de CodeCup competitie meedoet als met de D-opgave van de olympiade. Een e-mailtje naar contest@codecup.nl triggert ons mogelijk nog eerder! 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.

Voor meer informatie over dit spel: <https://en.wikipedia.org/wiki/Gomoku>.