

De eerste ronde Nederlandse Informatica Olympiade 2016-2017



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 juli 2017 in Iran.

De eerste ronde

De eerste ronde van de Nederlandse Informatica Olympiade bestaat dit jaar uit 11 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 opgave	1	200	200
D	Een spel programmeren	1	100	100

De beste 100 leerlingen worden uitgenodigd voor de tweede ronde, die in maart 2017 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

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 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 25 januari worden geupload. Op 28 januari wordt de eerste ronde gejureerd en kort daarna worden de uitslagen gepubliceerd.

Voor de spelopgave, opgave D, moet je je aanmelden op nio3.codecup.nl en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 28 januari 2016 tegen elkaar uit in een toernooi dat te volgen is op nio3.codecup.nl en de winnaar wint de jaarlijkse Windesheim Digitalisprijs van 200 euro.

Voor alle opgaven geldt dat je er van 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; zie de link op www.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 het goede antwoord instuurt krijg je 50 punten per opgave. Voor iedere dag later gaat er één punt van je score af. Inzendingen na 25 januari 2017 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, kunt oplossen.

Opgaven C1 tot en met C4

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. In één deelopgave hoef je alleen het antwoord op een vraag in te sturen.

Opgave D

Bij deze opgave moet je een programma schrijven dat het spel More kan spelen. De programma's spelen op 28 januari een toernooi tegen elkaar. Om deel te kunnen nemen moet je programma kunnen samenwerken met onze jurysoftware; voor details verwijzen we naar nio3.codecup.nl

De CodeCup

De Stichting Nederlandse Informatica Olympiade organiseert nog een wedstrijd waarbij een spel moet worden geprogrammeerd. Daaraan mag iedereen deelnemen; de afgelopen jaren hebben we deelnemers gehad uit meer dan twintig verschillende landen. Zie voor deze wedstrijd www.codecup.nl

Opgave A1. Vierkant

Schrijf een programma dat een getal N inleest van standard input. Het programma schrijft naar standard output een “vierkant” van N regels van elk N tekens, in een patroon zoals te zien is in het voorbeeld hieronder.

Voorbeeld

Invoer: 7

Uitvoer: * * * * *
 * - - - - *
 * - - - - *
 * - - - - *
 * - - - - *
 * - - - - *
 * - - - - *
 * * * * *

Randvoorwaarde: N is een positief geheel getal, met $0 < N < 41$

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave A2. Klnkrwg

Schrijf een programma dat een woord inleest van standard input. Het woord wordt geschreven met kleine letters. Bijzondere tekens als accenten, spaties en dergelijke worden niet gebruikt.

Het programma schrijft één regel uitvoer naar standard output. Op die regel staat het woord uit de invoer, maar alle klinkers worden weggelaten. (Klinkers zijn de letters a, e, i, o, u en y; de ij geldt als twee afzonderlijke letters, de i is wel een klinker, de j niet).

Voorbeeld:

Invoer: vergeetmijnietje

Uitvoer: vrgtmjntj

Randvoorwaarde: In de invoer staat een woord van maximaal 60 letters, waaronder minstens één medeklinker.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave A3. Collatz

Schrijf een programma dat van standard input een positief geheel getal N inleest. Je programma gaat telkens een nieuw getal N uitrekenen.

Afhankelijk van het getal N zijn er drie mogelijkheden.

- Als $N = 1$ is stopt het programma (en geeft het de hieronder gevraagde uitvoer).
- Als N even is, wordt de nieuwe N de helft van de oude. (Van 14 maakt het programma 7.)
- Als N oneven is, wordt de nieuwe N driemaal de oude N plus 1. (Van 7 naar 22.)

Je programma schrijft naar standard output de grootste waarde die het getal N heeft aangenomen.

Voorbeeld

Invoer: 14

Uitvoer: 52

Toelichting:

De hele rij getallen die je krijgt is

7	22	11	34	17	52	26	13	40	20	10	5	16
8	4	2	1									

Het vermoeden van Collatz (1937) zegt dat je op deze manier altijd op 1 zult uitkomen.

Dat klopt voor alle getallen van hoogstens achttien cijfers, maar is niet algemeen bewezen.

Randvoorwaarde: De invoer bestaat uit één getal, groter dan 1 en kleiner dan 1000000.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave A4. Vullen

1	3	4	6	2	4	2	3	3	1
1	2	4	3	2	2	2	4	4	5
1	4	4	2	3	2	3	4	5	5
4	4	6	5	5	5	5	5	5	4
4	3	5	1	5	2	5	1	5	3
2	1	5	5	5	1	5	5	3	3
5	5	5	5	5	5	5	5	5	5
5	2	5	6	5	4	5	3	5	4
1	6	2	3	2	3	3	2	6	6
6	6	1	3	4	1	2	4	5	5

Deze opgave gaat over een soort dambord van gekleurde vakjes. Vakjes die met een zijde aan een ander vakje met dezelfde kleur grenzen maken deel uit van hetzelfde gebied. Je kunt niet van het ene gebied naar het andere komen door horizontaal of verticaal één stapje te doen, zonder dat je daarbij van kleur verandert.

Het gebied waarin het vakje linksboven zit heet het basisgebied. Dit basisgebied krijgt telkens een nieuwe kleur. Op die manier wordt het gebied steeds groter, omdat aangrenzende vakjes met dezelfde kleur nu bij het gebied gaan horen. Als je dit blijft herhalen, is het hele bord uiteindelijk gevuld met vakjes van dezelfde kleur.

In iedere beurt zoek je van links naar rechts naar het eerste vakje dat een andere kleur heeft dan het basisgebied. Als je op het einde van de regel bent gekomen, zoek je door op de volgende regel. Met die kleur wordt dan het basisgebied gekleurd, zodat dit vakje wordt opgeslokt en deel gaat uitmaken van het basisgebied. Natuurlijk is het mogelijk dat dan nog meer vakjes deel gaan uitmaken van het basisgebied.

Schrijf een programma dat tien regels van tien tekens inleest van standard input. Iedere regel bestaat uitsluitend uit de cijfers van 1 tot en met 6. Die cijfers staan voor de verschillende kleuren. Het dambord heeft een grootte van 10x10 vakjes.

Je programma telt het aantal beurten dat nodig is om het hele dambord dezelfde kleur te geven volgens het hierboven beschreven voorschrift.

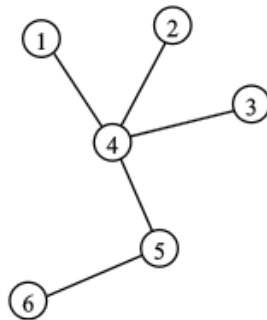
Voorbeeld

Invoer: 1346242331
1243222445
1442323455
446555554
4351525153
2155515533
555555555
5256545354
1623233266
6613412455

Uitvoer: 17

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave A5. Boomcentrum



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 bom met zes punten en vijf lijnen.

De afstand tussen twee punten in een boom wordt bepaald door het aantal lijnen dat je nodig hebt om van het ene punt naar het andere te komen. In de figuur is de afstand van punt 2 tot punt 6 daarom 3.

Het centrum van een boom wordt gevormd door de punten waarvan de grootste afstanden tot alle andere punten minimaal zijn. In de figuur hierboven wordt het centrum gevormd door de punten 4 en 5; de grootste afstand vanuit deze punten tot elk van de andere punten is 2.

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 het centrum van de boom uit de invoer bepalen. Het centrum 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 5
```

Randvoorwaarde: N is niet groter dan 150.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Kaartjes

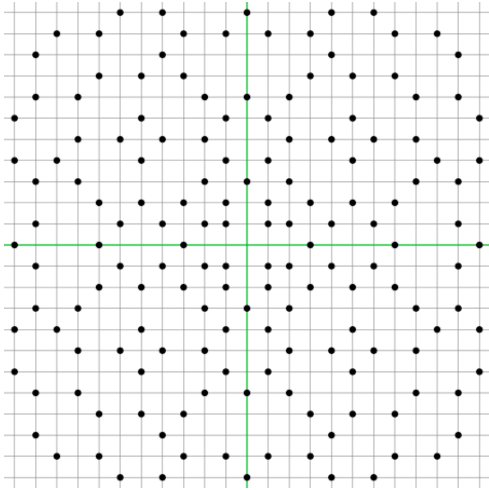
B2. Folders

B3. Meer

B4. Net

Opgave C. Priempunten en -lussen

Net zoals je op een x-as priemgetallen kunt weergeven, zo kan je dat ook doen in een plat vlak. De wiskundige Gauss heeft hier veel onderzoek aan gedaan en daarom zijn deze priemgetallen naar hem genoemd.

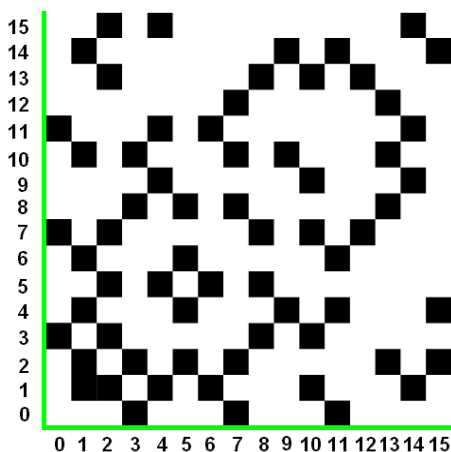


Het punt met de coördinaten (a, b) kan alleen een Gauss priemgetal zijn als a en b integers zijn.

- als a en b beiden niet 0 zijn, dan is (a, b) een Gauss priemgetal als $a^2 + b^2$ een priemgetal is
- als $a = 0$, dan is $(0, b)$ een Gauss priemgetal als $abs(b)$ een priemgetal is én $abs(b) \bmod 4 = 3$
- als $b = 0$, dan is $(a, 0)$ een Gauss priemgetal als $abs(a)$ een priemgetal is én $abs(a) \bmod 4 = 3$

Met $abs(x)$ wordt het getal x bedoeld als x al positief was en $-x$ als x negatief is. Met $y \bmod x = z$ wordt bedoeld dat de rest van de deling y / x het resultaat z heeft.

We beschouwen in deze opgaven alleen Gauss priemgetallen waarvan beide coördinaten positief zijn of nul. We kijken dus alleen naar situaties in het eerste kwadrant.



Opdracht C1: Gauss priemgetallen

Opdracht: lees vanaf standard input een reeks van tien coördinaten (a, b) in. Onderzoek van deze coördinaten of het Gauss priemgetallen zijn en schrijf naar standard output het resultaat weg. Gebruik hierbij de woorden "ja" of "nee".

Voorbeeld invoer:

```
1 1
2 2
4 5
4 6
4 7
7 13
8 4
12 7
14 7
15 14
```

Verwachte uitvoer:

```
ja
nee
ja
nee
nee
nee
nee
ja
nee
ja
```

Bij de beoordeling van je programma geldt het volgende:

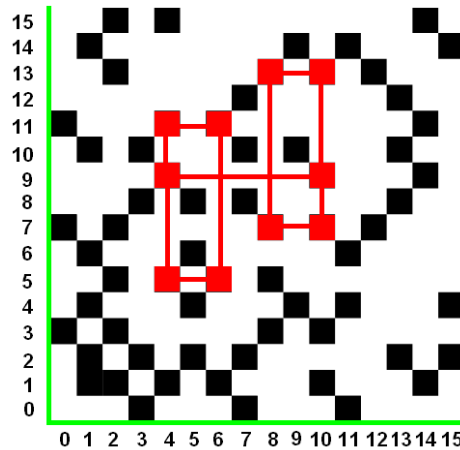
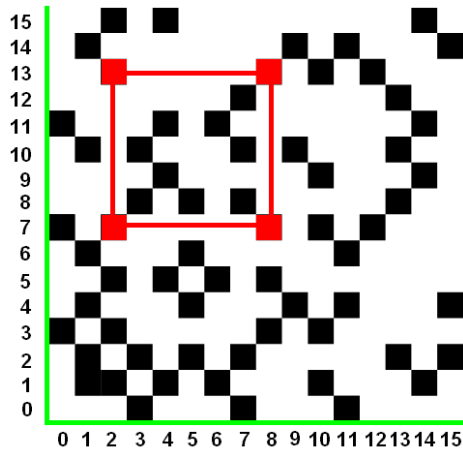
- De eerste test is 10 punten waard. Hierbij: $a \leq 60$ en $b \leq 60$.
- De tweede test is 10 punten waard. Hierbij: $a \leq 600$ en $b \leq 600$.
- De derde test is 10 punten waard. Hierbij: $a \leq 6000$ en $b \leq 6000$.

Je krijgt alleen punten voor de test als je programma alle uitvoer goed afhandelt.

Totaal te behalen voor C1: 30 punten

Opdracht C2: Lengte van een lus

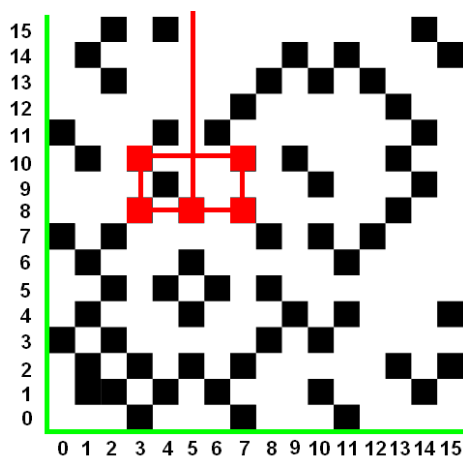
Het is mogelijk om in het eerste kwadrant lussen te maken. Hieronder zie je daarvan twee voorbeelden.



In het linker voorbeeld begin je bij de coördinaten (2,7). Je gaat vervolgens naar rechts totdat je op een nieuwe punt terecht komt. Dan ga je linksaf en ga je weer rechtdoor totdat je bij het volgende punt komt. Je gaat net zolang door met telkens linksaf slaan, totdat je weer op het oorspronkelijke punt uitkomt. In de rechter figuur zie je een andere oplossing, je kan starten bij (4,5).

In het linker voorbeeld is de lengte van de lus 24. In het rechter voorbeeld is de lengte 44.

We zijn pas tevreden met de lus als je bij de eerstvolgende beurt linksaf de lus weer gaat herhalen. Als je bij het herhalen de andere kant uitgaat en je niet meer op het startpunt terecht komt, is de lus dus ongeldig! Hieronder staat daarvan een voorbeeld. Het startpunt is (5,8).



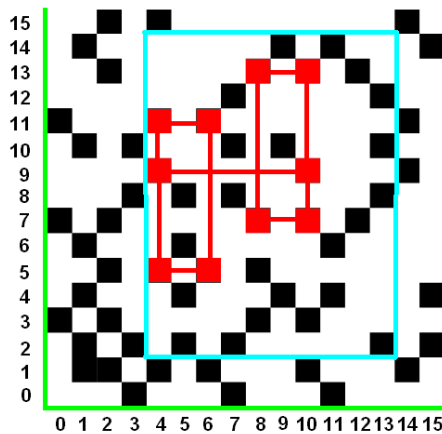
De lus moet volledig binnen het zoekgebied vallen. In het voorbeeld was het zoekgebied van (0,0) tot en met (15,15).

Opdracht: lees eerst vanaf standard input een zoekgebied in. Het zoekgebied loopt van (*links, onder*) tot en met (*rechts, boven*). De volgorde waarin de coördinaten gegeven worden is: *links, rechts, onder, boven*. Lees vervolgens de twee coördinaten (*a, b*) van een startpunt in. Schrijf naar standard output de lengte van de bijbehorende lus weg.

Bij deze opgave kan je er zeker van zijn dat je een geldige lus krijgt die in het zoekgebied valt.

Voorbeeld invoer:

```
4 13 2 14
4 9
```



Verwachte uitvoer:

```
44
```

Bij de beoordeling van je programma geldt het volgende:

- De eerste zes testen zijn elk 5 punten waard. Hierbij: $a \leq 60$ en $b \leq 60$. De zoekgebieden kunnen lopen van (0,0) tot en met (60,60).
- De volgende vier testen zijn elk 5 punten waard. Hierbij: $a \leq 600$ en $b \leq 600$. De zoekgebieden kunnen lopen van (0,0) tot en met (600,600).
- De laatste twee testen zijn elk 5 punten waard. Hierbij: $a \leq 6000$ en $b \leq 6000$. De zoekgebieden kunnen lopen van (0,0) tot en met (6000,6000).

Voor je programma geldt een tijdlimiet van 5 seconden.

Totaal te behalen voor C2: 60 punten

Opdracht C3: Langste lus

Deze opgave is een vervolg op opgave C2. Het is mogelijk om binnen het zoekgebied van $(0,0)$ tot en met $(6000,6000)$ een geldige lus te vinden met een maximale lengte. De optimale oplossing ziet er als volgt uit:



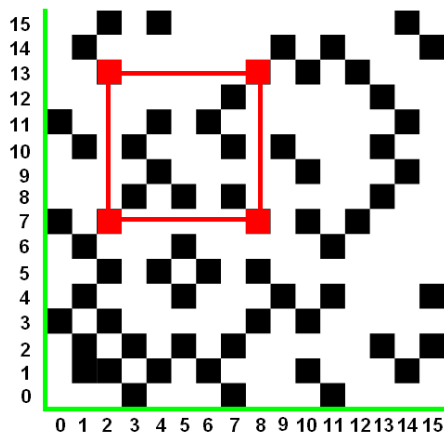
Opdracht: Zoek de lengte van de langste lus binnen het zoekgebied van $(0,0)$ tot en met $(6000,6000)$.

Lever een tekstbestand in met één regel, de lengte van de maximale lus. Bij deze opgave wordt pas na afloop bekend gemaakt of je gegeven antwoord juist is.

Totaal te behalen voor opgave C3: 20 punten.

Opdracht C4: Grootste vierkant

Deze opgave is ook een vervolg op opgave C2. Het zoekgebied bevat soms een lus dat tevens een vierkant is. Dit is daarvan een voorbeeld.



Er zijn geen vierkanten binnen het zoekgebied van (0,0) tot en met (6000,6000) te vinden met een breedte groter dan 30.

Opdracht: lees vanaf standard input een zoekgebied in. Het zoekgebied loopt van (*links, onder*) tot en met (*rechts, boven*). De volgorde waarin de coördinaten gegeven worden is: *links, rechts, onder, boven*. In dit zoekgebied is minstens één vierkant te vinden. Schrijf een programma dat het grootste vierkant binnen het zoekgebied zoekt (dus met de grootste lengte en breedte). Als er meer vierkanten met gelijke breedte gevonden worden, selecteer dan het vierkant waarvan het startpunt het dichtst bij de oorsprong ligt; gebruik hierbij de stelling van Pythagoras. Stel dat (a, b) het startpunt van het dichtst bijgelegen grootste vierkant is. Schrijf naar standard output de uitkomst weg van $a + b$.

Voorbeeld input:

```
0 10 5 15
```

Verwachte output:

```
9
```

Bij de beoordeling van je programma geldt het volgende:

- De eerste vier testen zijn elk 10 punten waard. Hierbij: $a \leq 60$ en $b \leq 60$. De zoekgebieden kunnen lopen van (0,0) tot en met (60,60).
- De volgende drie testen zijn elk 10 punten waard. Hierbij: $a \leq 600$ en $b \leq 600$. De zoekgebieden kunnen lopen van (0,0) tot en met (600,600).
- De laatste twee testen zijn elk 10 punten waard. Hierbij: $a \leq 6000$ en $b \leq 6000$. Voor de rechterbovenhoek van het zoekgebied geldt: $rechts \leq 6000$ en $boven \leq 6000$. De zoekgebieden hebben een maximale breedte en hoogte van 3000.

Totaal te behalen voor opgave C4: 90 punten.

Voor je programma geldt een tijdslimiet van 5 seconden.

Opgave D. More

Het spel

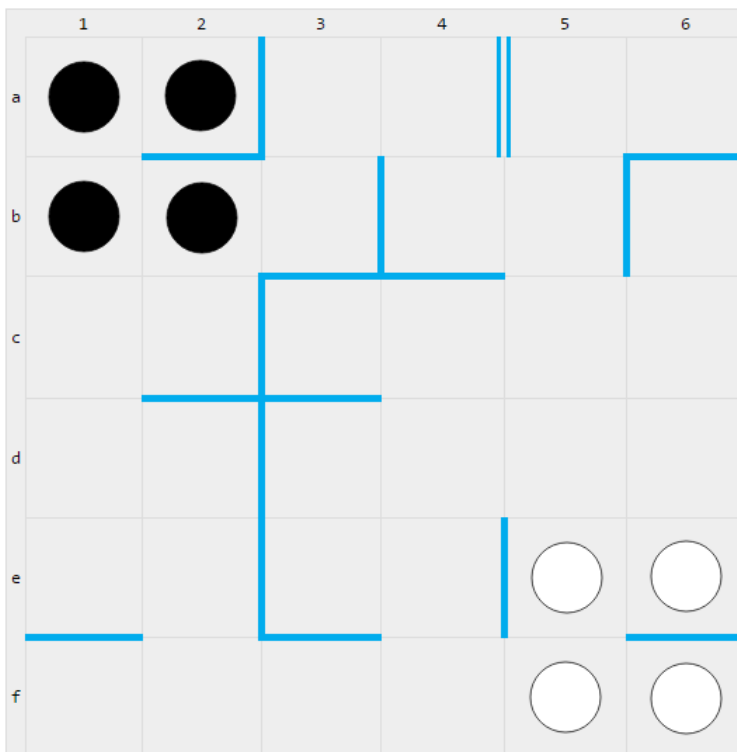
Het spel More is afgeleid van het spel Less (<http://www.less-game.com/>) dat op Kickstarter gestaan heeft. Voor More hebben we de regels zo aangepast dat het makkelijker te programmeren is. Het spel heet More omdat je meer zetten moet doen.

Het bord

More is een spel voor twee spelers en wordt gespeeld op een bord van 6 bij 6 hokjes. Tussen deze hokjes kunnen zich enkele of dubbele muurtjes bevinden. Deze hebben invloed op de score. De positie van alle muurtjes worden in het begin naar je programma gestuurd. Jouw programma moet aan het begin van het spel dit bord inlezen. Het protocol hiervoor staat onder het kopje IO. Dit bord wordt door het jurysysteem bij elk potje gegenereerd en zal dus steeds anders zijn, ook al speel je twee keer tegen dezelfde tegenstander. Bij het testen van je programma thuis is het mogelijk om steeds met hetzelfde bord te spelen. Dit wordt uitgelegd in de documentatie op nio3.codecup.nl bij het menu-item *Caia*.

Startpositie

Bij het begin van het spel heeft wit vier stenen rechts onderin het bord en heeft zwart vier stenen links bovenin het bord. Deze staan in een 2 bij 2 vierkantje.



Figuur 1 Een mogelijk bord bij het begin van het spel

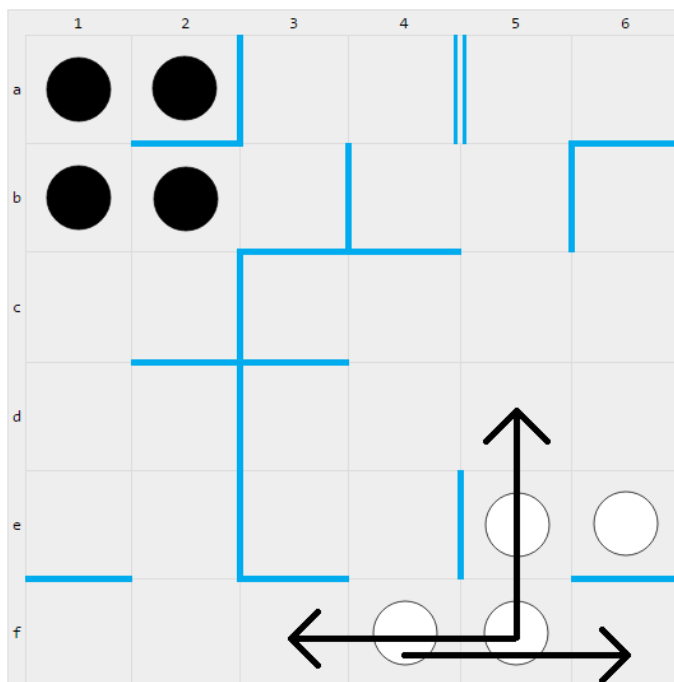
Regels

Wit begint met het spelen van een zet. Hierna doet zwart een zet. Dit blijft zo doorgaan tot een van de spelers al zijn stenen in de hoek van de andere speler heeft staan. Als de wedstrijd klaar is zal de score berekend worden.

Zet

In een beurt moet een speler precies één zet doen. Hierbij kan gekozen worden uit drie opties:

1. De speler kan een steen bewegen naar een aangrenzend hokje waar nog geen steen staat. Hierbij mag niet diagonaal bewogen worden.
2. De speler kan een steen bewegen naar een aangrenzend hokje waar nog geen steen staat, maar in dit geval bevindt zich een enkele muur of een dubbele muur ertussen. Als deze optie wordt gespeeld, zal er voor elk enkel muurtje één strafpunt aan het eind worden gerekend voor de speler en voor elk dubbel muurtje twee strafpunten.
3. De speler kan een steen verticaal of horizontaal over een andere steen bewegen naar een hokje dat 2 hokjes verder ligt. Voorwaarde is dat er geen muurtjes tussen zitten. Je kan niet diagonaal bewegen.



Figuur 2 Alle mogelijke sprongen over een andere steen voor wit

In en uitvoer

Jouw programma moet zich aan een protocol houden. De zetten van de jurysoftware moet je lezen vanaf stdin (standaard input) en je eigen zetten moet je afdrucken naar stdout (standaard uitvoer). Je bent wel vrij om tekst te sturen naar standard error (stderr).

Als eerste moet je het bord inlezen. Dit bord staat op de eerste regel van stdin. Het bord wordt beschreven door een string van 60 karakters. Elk karakter is een '0', een '1' of een '2'. Een '0' geeft aan dat er geen muur is, '1' wordt gebruikt voor een enkele muur en '2' voor een dubbele muur.

De muurtjes worden doorgegeven van links naar rechts en dan zo langzaam naar beneden werkend: eerst worden de 5 verticale muren op de eerste rij doorgegeven, daarna alle 6 horizontale muren tussen de eerste en de tweede rij. Daarna komen de 5 verticale muren op de tweede rij, enzovoort. Als laatste zijn de 5 verticale muren van de 6^{de} rij. Neem bijvoorbeeld voor de eerste 5 cijfers 00102, dan staat er een enkele muur tussen hokje a3 en a4, en een dubbele muur tussen a5 en a6. Als de 6 cijfers daarna bijvoorbeeld 100102 zijn, dan staat er een enkele muur tussen a1 en b1, en tussen a4 en b4, en een dubbele muur tussen a6 en b6. Het bord wat hoort bij figuur 1 is:

```
"010200100010010100110001000011000010000000000101010100100000"
```

Nadat het bord gestuurd is, krijgt je programma het woord "Start" (zonder de aanhalingstekens binnen) als je programma wit is. Als je programma zwart is, krijgt het de eerste zet door van de witte speler. De zet "a1a3" (zonder aanhalingstekens) houdt in dat de steen van plaats a1 naar a3 verplaatst wordt.

In beide gevallen moet je antwoorden met de zet die jij wilt doen (in het geval dat je programma wit is, is dit de eerste zet). Hierna krijg je de zet die de andere speler doet, en moet je weer een zet geven. Dit blijft doorgaan tot je programma "Quit" binnenkrijgt. Als je programma "Quit" inleest in plaats van een gewone zet, dan betekent het dat het spel is afgelopen en dat je jouw programma op de normale manier moet beëindigen.

Als één van de twee programma's klaar is wordt de wedstrijd afgemaakt totdat ook het andere programma al zijn stenen in de hoek heeft staan. Het programma die nog steeds bezig is krijgt steeds "Nil" ingelezen om aan te geven dat het andere programma al klaar is. Dus als je programma "Nil" krijgt ingelezen moet je meteen beginnen met het berekenen van je volgende zet en deze teruggeven.

Als je een zet doorgeeft aan de jurysoftware moet je de string verplicht afsluiten met een newline. Ook ben je verplicht om je uitvoer daarna te flushen zodat de jurysoftware niet gaat zitten wachten totdat de computer de zet heeft doorgegeven aan de jurysoftware (zie de uitleg bij de *technische regels*).

Score

Aan het eind van het spel wordt de score berekend. Het puntenaantal is de som van het aantal zetten en het aantal strafpunten. Jouw score is het puntenaantal van je tegenstander min jouw puntenaantal. Hier komen altijd nog 10 punten bij. Als je score kleiner dan 0 is, krijg je als score 0. Als je score groter dan 20 is, krijg je als score 20. Bij een gelijkspel hebben beide spelers dus de score 10. De som van beide scores is zo ook altijd 20.

Voorbeeld: Jij bent aan de overkant in 55 zetten en je hebt 4 strafpunten gekregen. Je tegenstander heeft er 60 zetten voor nodig gehad en kreeg 2 strafpunten. Jouw puntenaantal is dan $55 + 4 = 59$, en je tegenstander heeft $60 + 2$ punten. Jouw score is dan $62 - 59 + 10 = 13$. Je tegenstander krijgt de score $59 - 62 + 10 = 7$.

Als je programma op enig moment van de wedstrijd crasht, een verkeerde zet teruggeeft of in totaal meer dan 30 seconden gebruikt voor één spel, dan krijgt je programma als score 0. De jury zal een vervangende speler inzetten die namens jou zetten doet, zodat de tegenstander nog verder kan spelen. Je tegenstander krijgt dan nog wel een score zoals het normaal zou gaan, maar jouw programma krijgt dan 0 punten.

Indien er in een spel meer dan 80 zetten zijn gespeeld, zal de jury software het spel afkappen. De jury zal voor beide spelers apart berekenen wat het kleinste aantal zetten is waarin een speler in de andere hoek kan komen zonder dat er stenen van de tegenstander op het bord staan.

Met alle ingezonden programma's die in staat zijn samen te werken met onze jurysoftware wordt een toernooi georganiseerd.

Communicatievoorbeeld

Invoer wit	Uitvoer wit	Invoer zwart	Uitvoer zwart
0102001000100101001100 0100001100001000000000 0101010100100000		01020010001001010011000 10000110000100000000001 01010100100000	
Start	f6f4		
		f6f4	a1c1
a1c1	f5d5		
		f5d5	b1d1
b1d1	e5c5		
		e5c5	c1e1
...	...		
	
...	b3b2		
		b3b2	...
		Nil	f4f6
		Nil	e4e5
Quit			
		Quit	

Je programma heeft 30 seconden de tijd om een spel te spelen. De tijd die de tegenstander heeft wordt niet meegeteld.

Deelnemen?

Meld je aan op nio3.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 winnaar 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.