

De eerste ronde Nederlandse Informatica Olympiade 2018-2019



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 2019 in Azerbeidzjan.

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 2019 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.

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 16 januari worden geüpload. Op 19 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 en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 19 januari 2019 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 19 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. 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 16 januari 2018 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 Flippo kan spelen. Aan dit toernooi doen ook andere deelnemers me, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 19 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. Trapezium

Schrijf een programma dat drie getallen N, M en V inleest, elk van een regel van standard input. Het programma schrijft naar standard output een rechthoekig trapezium. De eerste regel bestaat uit N keer een *, bij elke volgende regel wordt het aantal * V groter, zoals te zien is in het voorbeeld hieronder. Je voert in totaal M regels uit.

Voorbeeld

Invoer: 5
 7
 2

Uitvoer: * * * * *
 * * * * * * *
 * * * * * * * * *
 * * * * * * * * * *
 * * * * * * * * * * *
 * * * * * * * * * * * *
 * * * * * * * * * * * * *
 * * * * * * * * * * * * * *

Toelichting:

- De eerste regel bestaat uit 5 keer een * (N = 5)
- Je ziet 7 regels (M = 7)
- Iedere regel bevat 2 sterretjes meer dan de voorafgaande (V = 2)

Randvoorwaarde:

N, M, en V zijn positieve gehele getallen, met $0 < N < 24$, $0 < M < 60$ en $0 < V < 5$.

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A2. Letterbeeld

De hoofdletter A heeft 1 gesloten gebied en 2 losse uiteinden. Ook andere hoofdletters kunnen gesloten gebieden omvatten (de B zelfs 2), en losse uiteinden hebben.

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.

Het programma schrijft twee regels uitvoer naar standard output. Op de eerste regel staat het aantal gesloten gebieden in de letters van het woord uit de invoer; op de tweede regel staat het aantal losse uiteinden van de letters van het woord.

Voorbeeld:

Invoer: **COMPUTER**

Uitvoer: 3
 15

Randvoorwaarde: In de invoer staat een woord van maximaal 60 letters.

Let op: we gaan uit van zogenaamde schreefloze letters, dus letters zonder dwarsstreepjes aan een uiteinde. Het is dus geen **C** maar een **C** en geen **I** maar een **I**. Hieronder zie je het alfabet voor de volledigheid:

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

Dus de **I** heeft twee uiteindes, net als de **Q**.

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A3. Cijferwoorden

In sommige woorden kun je als je een deel van de letters weglaat een cijfer herkennen.

VERD~~AC~~HT VERD~~AC~~HT ACHT

VERDRI~~E~~T VERDRI~~E~~T DRIE

VERTI~~E~~R VERTI~~E~~R VIER

Omdat EEN een kort woord is dat je vaak zult kunnen herkennen tellen we dat hier niet mee. Een woord is een cijferwoord als je door één of meer letters weg te laten precies één van de woorden TWEE, DRIE, VIER, VIJF, ZES, ZEVEN, ACHT of NEGEN kunt maken.

Schrijf een programma dat van standard input één woord inleest, geschreven in uitsluitend hoofdletters. Je programma schrijft naar standaard output één regel, met daarop als er een cijfernaam te vinden is in het woord van de invoer de naam van dat cijfer, uitgeschreven in hoofdletters. Als er geen cijfernaam te vinden is schrijft je programma het woord `geen`.

Voorbeeld

Invoer: ZWERVERS

Uitvoer: ZES

Het woord in de invoer bestaat uit minstens 5 en hoogstens 20 letters.

Er zijn geen testgevallen waar in het woord van de invoer meer dan één cijfernaam kan worden gevonden.

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A4. Getallenrivieren

Bij ieder positief natuurlijk getal N kun je een getallenrivier maken. Je begint met N , neem bijvoorbeeld $N = 2019$, en telt om het volgende getal te krijgen de waarde van elk van de cijfers bij het getal op. Dus het volgende getal wordt $2019 + 2 + 0 + 1 + 9 = 2031$. En dat kun je doorzetten; zo krijg je een oneindige oplopende rij getallen, de rivier van 2019.

2019
2031
2037
2049
2064
2076
Etc....

Soms komen rivieren samen, zoals de Moezel en de Rijn, of de Regge en de Vecht. De rivier van 79 gaat verder met 95, 109, 119 en de rivier van 80 gaat verder met 88, 104, 109, 119 en dus stromen ze vanaf 109 samen.

Er zijn drie grote rivieren in getallenland, de rivieren van 1, 3 en 9. Iedere rivier blijkt na een aanloop samen te stromen met één van deze drie rivieren. Zo gaan de rivieren van 79 en 80 vanaf 620 samen met de rivier van 1.

Schrijf een programma dat van standaard input eerst een getal N inleest ($1 < N < 16500$).

Je programma geeft als uitvoer twee regels naar standaard output; op de eerste regel staat een 1, 3 of 9 om aan te geven dat de rivier van N met deze grote rivier gaat samenstromen. Op de tweede regel staat het getal vanaf waar dat het geval is. Dat tweede getal is zeker kleiner dan 1000000 (een miljoen).

Voorbeeld

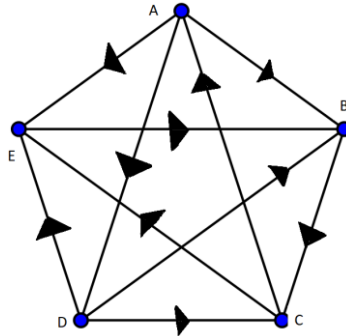
Invoer: 44

Uitvoer: 1
 107

Voor je programma geldt een tijdslimiet van 2 seconden.

Opgave A5. Eénrichtingsverkeer

In een bepaalde regio zijn alle plaatsen door een rechtstreekse weg met elkaar verbonden. Maar voor alle wegen geldt éénrichtingsverkeer; als je van A naar B mag via de directe weg, mag je dus niet langs die weg van B naar A. Als je dat wel wilt zul je een omweg moeten nemen.



Hierboven een voorbeeld van een gebied met vijf plaatsen en tussen ieder paar plaatsen een weg die je maar één kant op mag doorlopen. De wegen CA en DB lijken elkaar te kruisen, maar het is daar niet mogelijk om van weg te switchen; ze lopen bijvoorbeeld onder en over een viaduct.

In zo'n gebied ga je op zoek naar een route waarbij je iedere plaats precies één keer bezoekt. In dit voorbeeld is zo'n route ADBCE of DCAEB of DBCAE. Let op: Er wordt niet gevraagd om nu weer terug te gaan naar de plaats waar je begonnen bent.

Schrijf een programma dat van standaard input eerst een getal N inleest ($2 < N < 27$).

Hiermee wordt het aantal plaatsen in het gebied aangegeven, de plaatsen hebben als naam een hoofdletter; de eerste N hoofdletters worden gebruikt als plaatsnaam.

Vervolgens leest je programma van standaard input in N regels informatie over de richting van de verbindingswegen. Op de eerste regel staan de N verbindingen van A naar alle andere plaatsen; een 0 betekent dat de weg in A eindigt, een 1 dat de weg in A begint. De volgende regels gaan over de volgende plaatsen. Tussen twee plaatsen is altijd precies één weg; er zijn geen wegen van een plaats naar zichzelf!

Je programma schrijft naar standaard output één regel met daarop een route waarin alle plaatsen precies één keer worden bezocht. Er is altijd een oplossing mogelijk.

Voorbeeld:

```
Invoer: 5
        01011
        00100
        10001
        01101
        01000
```

```
Uitvoer: ADBCE
```


Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Viercijferwoord

B2. Concentrië

B3. Rooster

B4. Van 1 naar ...

Opgave C1. Passen

Met euromunten kun je op allerlei manieren een bedrag gepast betalen.



Als je 13 cent moet betalen, kun je, als je tenminste van alle munten genoeg op voorraad hebt, al kiezen uit zestien verschillende manieren:

| Manier | Aantal van 10 | Aantal van 5 | Aantal van 2 | Aantal van 1 |
|--------|---------------|--------------|--------------|--------------|
| 1 | 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 0 | 3 |
| 3 | 0 | 2 | 1 | 1 |
| 4 | 0 | 2 | 0 | 3 |
| 5 | 0 | 1 | 4 | 0 |
| 6 | 0 | 1 | 3 | 2 |
| 7 | 0 | 1 | 2 | 4 |
| 8 | 0 | 1 | 1 | 6 |
| 9 | 0 | 1 | 0 | 8 |
| 10 | 0 | 0 | 6 | 1 |
| 11 | 0 | 0 | 5 | 3 |
| 12 | 0 | 0 | 4 | 5 |
| 13 | 0 | 0 | 3 | 7 |
| 14 | 0 | 0 | 2 | 9 |
| 15 | 0 | 0 | 1 | 11 |
| 16 | 0 | 0 | 0 | 13 |

Er zijn ook wel andere muntsystemen. Als je alleen muntjes van 1, 13 of 169 cent hebt, kun je 13 cent maar op 2 manieren gepast betalen (ga dat zelf na).

In deze opgave krijg je te zien welke soorten munten je mag gebruiken en welk bedrag je moet betalen. Jouw programma moet uitrekenen op hoeveel verschillende manieren dat kan. Let op, dat kan gaan om een heel groot getal!

Schrijf een programma dat de gegevens over beschikbare munten en een te betalen bedrag inleest van standard input.

Op de eerste regel staat een getal N , dat aangeeft er N verschillende muntsoorten bestaan. Op de volgende N regels staan de waardes van die munten, telkens een geheel getal, in oplopende volgorde. Daarna volgt een regel met het bedrag B . Er geldt $0 < N < 9$, de waardes van de verschillende munten worden niet groter dan 250 en voor het bedrag dat moet worden betaald geldt

$0 < B < 300$. Bij de beschikbare munten zit altijd een munt met waarde 1, zodat je ieder bedrag altijd kunt betalen.

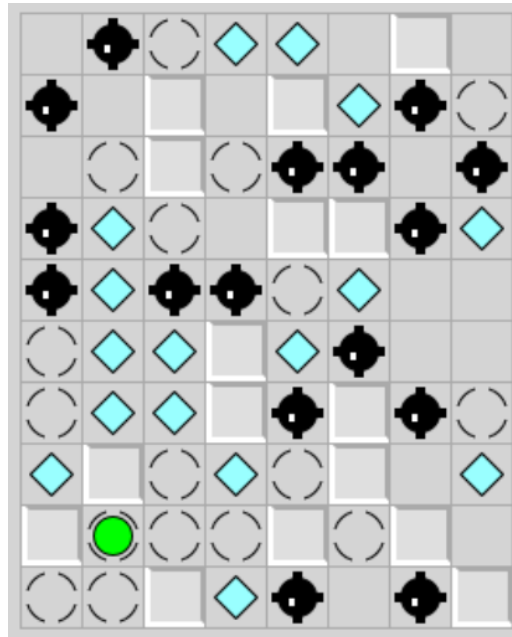
Je programma moet bepalen op hoeveel verschillende manieren bedrag B kan worden betaald en voert dat aantal uit naar standard output.

Voorbeeld:

| | |
|---------|----|
| Invoer | 4 |
| | 1 |
| | 2 |
| | 5 |
| | 10 |
| | 13 |
| Uitvoer | 16 |

Voor je programma geldt een tijdslimiet van 5 seconden.

Opgave C2. Traagheid



In het spel Traagheid stuur je als speler een groene bal in een rooster vol obstakels. Jij moet alle edelstenen verzamelen zonder op een mijn te lopen.

Je kunt je bal in acht richtingen bewegen, horizontaal, verticaal of diagonaal. Als de bal eenmaal beweegt, blijft hij bewegen tot hij wordt tegengehouden. Als hij een muur op zijn weg vindt zal hij stoppen (maar als hij diagonaal beweegt zal hij door een gat tussen twee diagonaal geplaatste muren bewegen zonder te worden gestopt). Op sommige vakjes is een stopplaats, de bal blijft daar liggen onafhankelijk van de richting waarin hij werd bewogen. Edelstenen houden de bal niet tegen; ze worden meegenomen en de bal blijft in dezelfde richting bewegen.

Als je tegen een mijn aanloopt is het spel voorbij. Zelfs als je de laatste edelsteen eerst oppakt en in dezelfde beweging daarna tegen een mijn aan loopt, heb je verloren.

Voor deze opgave schrijf je een programma dat een beginpositie van Traagheid inleest. Jouw programma voert een serie aanwijzingen uit waarmee het spel bij deze beginpositie kan worden gespeeld.

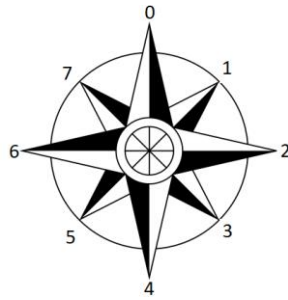
Je programma leest van standaard input 10 regels met elk 8 tekens. De betekenis van deze tekens is:

| | | |
|---|-------------------------|---|
| @ | Beginpositie van de bal | (1 vakje, dit is ook een stopplaats) |
| + | Edelsteen | (16 vakjes) |
| # | Muur | (16 vakjes) |
| * | Bom | (16 vakjes) |
| o | Stopplaats | (16 vakjes) (let op, dit is de hoofdletter o) |
| - | Vrije ruimte | (15 vakjes) |

Je programma schrijft naar standaard output één regel met daarop een serie aanwijzingen in de vorm van cijfers. De cijfers van 0 tot en met 7 worden gebruikt om aan te geven in welke richting de

bal moet worden bewogen; voor een betekenis zie de windroos hieronder. Je programma moet aan de volgende eisen voldoen:

1. Het programma moet de bal zoveel mogelijk edelstenen laten meenemen.
2. De bal mag nooit op een bom terecht komen.
3. De bal mag maximaal 20 keer achter elkaar worden bewogen zonder een edelsteen mee te kunnen nemen.



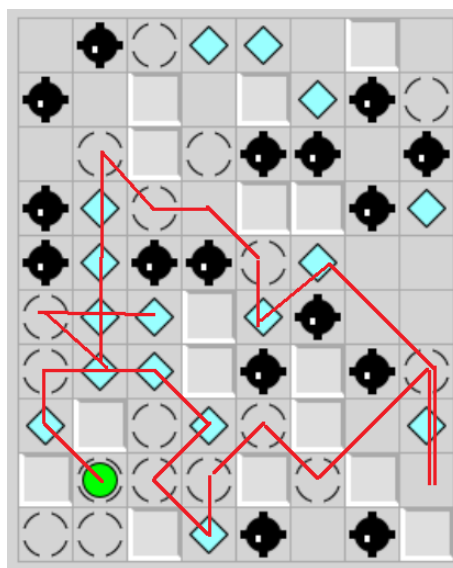
Voorbeeld:

Invoer:

```

- * O + + - # -
* - # - # + * O
- O # O * * - *
* + O - # # * +
* + * * O + - -
O + + # + * - -
O + + # * # * O
+ # O + O # - +
# @ O O # O # -
O O # + * - * #
    
```

Uitvoer: 702353013140750767472



De route bij de voorbeelduitvoer





Op de aangegeven manier kun je 12 van de 16 edelstenen oppikken. Er is nog meer mogelijk.

Je programma moet vijf testgevallen oplossen. In alle gevallen is het mogelijk alle 16 edelstenen op te pikken. Als je bij een testgeval dit maximale aantal edelstenen hebt meegenomen krijg je in ieder geval 16 punten voor die test; anders kost iedere edelsteen die je gemist hebt 2 punten. Als je programma het testgeval goed heeft opgelost en het ook nog eens een beperkt aantal bewegingen nodig had, kan het nog maximaal 4 punten extra krijgen voor die test.

De laatste vier punten per testgeval worden pas op de jurydag toegekend; voor die tijd is de maximaal haalbare score voor deze opgave 80 punten.

Opgave D. Flippo

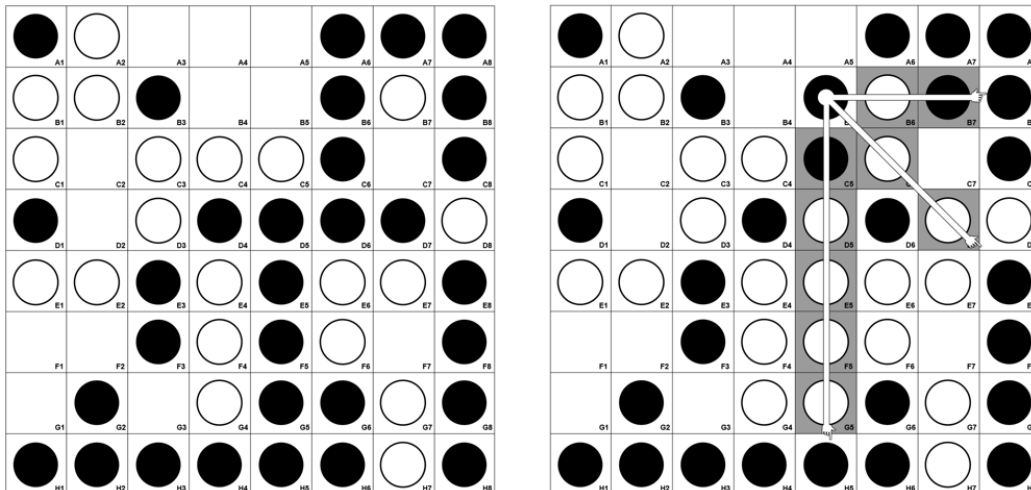
Het spel Flippo is een alternatief op het bekende spel Reversi of Othello. Flippo wordt gespeeld met stenen die een zwarte en witte kant hebben, op een vierkant bord van 8 bij 8 vakjes. De beginpositie zie je hieronder:

| | | | | | | | |
|----|----|----|--|--|----|----|----|
| | | | | | | | |
| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 |
| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
| D1 | D2 | D3 |  |  | D6 | D7 | D8 |
| E1 | E2 | E3 |  |  | E6 | E7 | E8 |
| F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 |
| G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 |
| H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 |

Spelverloop

De speler met wit begint. Daarna zetten de spelers om beurten een steen op het bord, met de kleur waarmee zij spelen naar boven. Het spel is voorbij als er 60 zetten zijn gedaan en het bord volledig gevuld is. Een speler moet één steen naast een rij stenen op het bord plaatsen, verticaal, horizontaal of diagonaal, zodat tussen de nieuwe steen en de steen van dezelfde kleur in de verticale, horizontale of diagonale rij die zo ver mogelijk van deze nieuwe steen af ligt tenminste één steen wordt ingesloten. Alle zo ingesloten stenen worden omgekeerd, waardoor ze van kleur veranderen. Als het niet lukt om een steen zo te plaatsen, moet deze aanliggend aan een groep stenen op het bord worden gelegd.

Voorbeeld van een zet



Bekijk de stand in het linker plaatje:

De speler met zwart speelt een zwarte steen op B5. Daarna flippen er een heleboel stenen; de witte stenen zijn zwart geworden en andersom.

Score

De uitslag van het spel hangt af van het aantal stenen van een kleur dat op het bord staat aan het einde van het spel. Als er W witte en Z zwarte stenen zijn, ontvangt de witte speler $W-2$ punten en de zwarte speler $Z-2$. De som van de zo behaalde punten is 60.

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

Als je programma de tijdslimiet overschrijdt of een onreglementaire zet doet, zoals het plaatsen van een steen op een vakje dat 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 op de wedstrijdwebsite 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 standaard input. En 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 wedstrijdwebsite.

De witte speler krijgt aan het begin van het spel 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 zwarte speler krijgt aan het begin van het spel 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 ziet eruit als "D3". De zet "D3" plaatst een steen in de kleur van de speler die aan de beurt is op vakje D3.

Voorbeeld van communicatie:

| Invoer Wit | Uitvoer Wit | Invoer Zwart | Uitvoer Zwart | Toelichting |
|------------|-------------|--------------|---------------|---------------|
| Start | | | | Wit begint |
| | D6 | D6 | | Zet van wit |
| C6 | | | C6 | Zet van zwart |
| | E3 | E3 | | Zet van wit |

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 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.