

De eerste ronde Nederlandse Informatica Olympiade 2024-2025



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 2025 in Bolivia.

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 349 punten dan staat op het Certificaat de vermelding **Brons**, tussen de 350 en 499 punten de vermelding **Zilver** en bij 500 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	25	100
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 2025 wordt gehouden op de Universiteit Twente. Voor deelname aan de tweede ronde moet je wel in totaal minstens 250 punten hebben gehaald

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 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 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 2025 tegen elkaar uit in een toernooi dat te volgen is op www.codecup.nl. Inzenden mag tot 18 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.

De programma's die je moet schrijven 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. Tip: je mag absoluut geen extra teksten afdrukken zoals 'geef je invoer' of 'het antwoord is'. Het jurysysteem is heel streng en rekent extra teksten altijd fout.

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. Ook is er een introductiecursus beschikbaar die samen met EGOI (European Girls' Olympiad in Informatics) is ontwikkeld en door iedereen gebruikt mag worden: <https://girls.gitbook.io/c++-cursus>

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 25 punten per opgave. Voor iedere dag later gaat er één punt van je score af. Inzendingen na 15 januari 2025 zullen niet worden verwerkt.

Als je een verkeerd antwoord hebt gegeven, verlies je meteen 5 punten, totdat er van de 25 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 direct 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 erop lijken, kunt oplossen.

Opgaven C1 en C2

Dit zijn complexere opgaven waarmee je een probleem moet oplossen door het schrijven van een computerprogramma.

Opgave D en de CodeCup

Bij deze opgave moet je een programma schrijven dat het spel Squares kan spelen. Dat is een variant van het spel Box van de CodeCup 2025. Aan het toernooi om de CodeCup doen ook andere deelnemers mee, soms wel uit meer dan twintig verschillende landen. Zie hiervoor ook www.codecup.nl

De programma's voor het spel Squares 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 nio.codecup.nl

Opgave A1. Middelste letter(s)

Schrijf een programma dat van standard input een regel met daarop een woord inleest.

Jouw programma schrijft naar standard output de middelste letter of twee letters van het woord in de invoer. Voor de middelste letter of letters en erna telt het woord evenveel letters.

Voorbeelden

Invoer:
AARDAPPEL

Invoer:
Aquarium

Invoer:
ER

Uitvoer:
A

Uitvoer:
ar

Uitvoer:
ER

De woorden in de invoer bestaan uit minstens 2 en hoogstens 80 letters. Dat zijn alleen hoofdletters (A..Z) en kleine letters (a..z), dus geen spaties, leestekens, cijfers, etc. en ook geen letters met accenten. De IJ als in BIJZONDER telt als twee letters.

Je moet de hoofd- en kleine letters van de invoer in stand houden in de uitvoer. Bij het eerste voorbeeld wordt 'a' dus niet goed gerekend!

Voor je programma geldt een tijdslimiet van 1 seconde.

Opgave A2. Tijdrit

Aan het begin en aan het einde van een tijdrit wordt de tijd vastgelegd in honderdsten van seconden nauwkeurig. Jouw programma moet de totale tijd tussen vertrektijd en eindtijd berekenen in uren, minuten, seconden en hondersten van seconden waarbij het aantal minuten en het aantal seconden nooit groter dan 60 is.

De tijd heeft de vorm `uu:mm:ss, hh` dus bijvoorbeeld `12:34:56,78`.

Voor een totale tijd onder de 10 uur wordt maar 1 teken gebruikt voor de uren, zoals `2:34:56,78`. Als de totale tijd minder dan een uur is vervallen de uren en de eerste dubbele punt.

Het aantal minuten is kleiner dan 60. Voor een totale tijd onder de 10 minuten wordt maar 1 teken gebruikt voor de minuten, zoals `3:45,67`. Als de totale tijd minder dan een minuut is vervallen ook de minuten en de tweede dubbele punt.

Het aantal seconden is kleiner dan 60. Voor een totale tijd onder de 10 seconden wordt maar 1 teken gebruikt voor de seconden, zoals `5,67`. Als de totale tijd minder dan een seconde is wordt dat aangegeven met een 0 voor de komma, zoals `0,67`

Schrijf een programma dat twee regels met tekst leest van standard input. Die tekst bestaat uit een regel met een begintijd in het aangegeven formaat, gevolgd door een regel met een eindtijd.

Het programma schrijft naar standard output één regel met daarin het tijdsverschil. Het is gegarandeerd dat de eindtijd na de begintijd is.

Voorbeelden

Invoer:
`12:34:56,78`
`14:01:01,01`

Invoer:
`9:58:47,36`
`10:04:12,99`

Invoer:
`0:59:59,90`
`1:00:00,87`

Uitvoer:
`1:26:04,23`

Uitvoer:
`5:25,63`

Uitvoer:
`0,97`

Voor je programma geldt een tijdslimiet van 1 seconde.

Opgave A3. NIO-limieten

In deze opgave wordt van een positief geheel getal zijn NIO-limiet bepaald. Dat gaat als volgt:

Van een positief geheel getal wordt het NIO-nummer berekend. Dat NIO-nummer krijg je door de cijfers uit het getal op een gewogen wijze bij elkaar op te tellen: 1 maal het eerste cijfer, 2 maal het tweede cijfer, 3 maal het derde cijfer, etc.

Het NIO-nummer van 2025 is $1 \cdot 2 + 2 \cdot 0 + 3 \cdot 2 + 4 \cdot 5 = 28$.

Dit berekende NIO-nummer is natuurlijk weer een positief geheel getal waar ook weer het NIO-nummer van bepaald kan worden

Het NIO-nummer van 28 is $1 \cdot 2 + 2 \cdot 8 = 18$.

Dit kun je herhalen totdat je telkens hetzelfde getal krijgt. Dit getal is de NIO-limiet van het positief gehele getal waar je mee begonnen bent.

Getal	NIO-nummer	Berekening
2025	28	$1 \cdot 2 + 2 \cdot 0 + 3 \cdot 2 + 4 \cdot 5 = 28$
28	18	$1 \cdot 2 + 2 \cdot 8$
18	17	$1 \cdot 1 + 2 \cdot 8$
17	15	$1 \cdot 1 + 2 \cdot 7$
15	11	$1 \cdot 1 + 2 \cdot 5$
11	3	$1 \cdot 1 + 2 \cdot 1$
3	3	NIO-limiet

In dit voorbeeld zeggen we dat de NIO-limiet van 2025 gelijk is aan 3.

Schrijf een programma dat een regel leest van standard input. Op die regel staat een getal N. Het programma schrijft één regel uitvoer naar standard output. Daarop staat het getal L dat de NIO-limiet is van N.

Voorbeelden:

Invoer:
159

Invoer:
132

Invoer:
3672108

Uitvoer:
19

Uitvoer:
7

Uitvoer:
7

Randvoorwaarde:

Voor de helft van de testgevallen geldt $1 \leq N \leq 1000$

Voor de overige testgevallen geldt $1000 \leq N \leq 1000000000$

Voor je programma geldt een tijdslimiet van 1 seconde.

Opgave A4. Componenten

Een graaf is een structuur met punten en lijnen.

De punten hebben een uniek nummer; er zijn N punten en ze worden genummerd van 1 tot en met N .

Er zijn M lijnen. Elke lijn verbindt twee verschillende punten met elkaar. Tussen twee punten is maximaal één lijn.

Een samenhangende deelgraaf is een gedeelte van de punten in de graaf die met lijnen zijn verbonden waardoor je via de lijnen vanuit elk van die punten naar elk ander punt kunt in dat deel.

Een component is een verzameling punten die allemaal onderling verbonden zijn, een zogenaamde maximale samenhangende deelgraaf. Een punt dat niet met andere punten verbonden is is dus in zijn eentje een component.

Schrijf een programma dat van standaard input

- eerst een getal N en een getal M inleest ($0 < N \leq 40\ 000$, $0 < M \leq 40\ 000$),
gescheiden door een spatie.

- vervolgens M regels inleest met elk twee getallen gescheiden door een spatie, dat zijn de punten die verbonden zijn door een lijn.

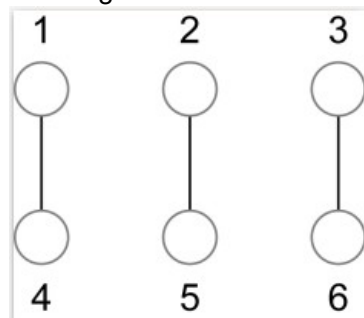
Je programma geeft als uitvoer één regel naar standaard output met daarop het aantal componenten van de graaf uit de invoer.

Voorbeelden

Invoer:

```
6 2  
1 4  
2 5
```

invoer grafisch:



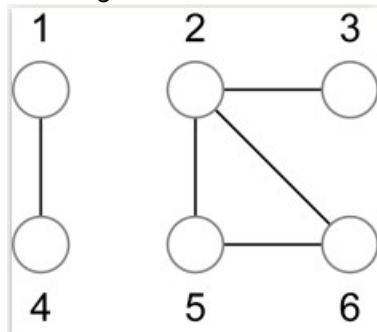
Uitvoer:

```
4
```


Invoer:

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

invoer grafisch:



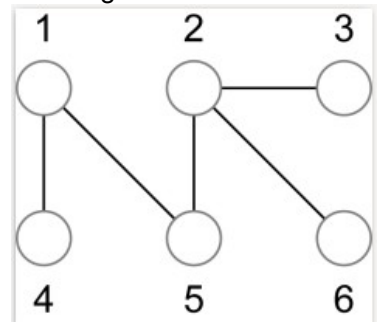
Uitvoer:

2

Invoer:

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

invoer grafisch:



Uitvoer:

1

Voor je programma geldt een tijdslimiet van 3 seconden.

Voor de testgevallen geldt het volgende:

- 5 testgevallen met $N < 6$
- 5 testgevallen met $N < 100$
- 10 testgevallen met $100 < N \leq 40\ 000$

Opgave A5. Schudden

Voor een kaartspelletje moet je de kaarten eerst schudden. Maar hoe wordt de volgorde na het schudden van kaarten in een stapel? In deze opgave doen we dat proces van schudden na volgens het hieronder beschreven model.

We hebben $2N$ genummerde kaarten op een stapel. Van boven naar onder ligt de kaart met nummer 1 boven, daaronder nummers 2, 3 en zo door tot onderaan kaart $2N$.

Als voorbeeld gebruiken we $N=4$, dan liggen de kaarten in de volgorde 1, 2, 3, 4, 5, 6, 7, 8.

1 _____
 2 _____
 3 _____
 4 _____
 5 _____
 6 _____
 7 _____
 8 _____

Er zijn drie bewerkingen die we met de stapel kunnen uitvoeren.

De rits-acties. Daarbij wordt de stapel in twee gelijke delen verdeeld, de bovenste N kaarten en de onderste N . In beide gevallen worden de kaarten in elkaar geritst, om en om een kaart van de ene en van de andere stapel.

de break (b)

de bovenste kaart wordt onderaan de stapel gelegd

2 _____
 3 _____
 4 _____
 5 _____
 6 _____
 7 _____
 8 _____
 _____ 1

de volgorde is nu

2, 3, 4, 5, 6, 7, 8, 1.

de rechter-rits (r)

de kaart die oorspronkelijk bovenop lag blijft bovenop de stapel

1 _____
 _____ 5
 2 _____
 _____ 6
 3 _____
 _____ 7
 4 _____
 _____ 8

de volgorde is nu

1, 5, 2, 6, 3, 7, 4, 8.

de linker-rits (l)

de kaart die oorspronkelijk bovenop lag ligt nu als tweede kaart op de stapel

5 _____
 _____ 1
 6 _____
 _____ 2
 7 _____
 _____ 3
 8 _____
 _____ 4

de volgorde is nu

5, 1, 6, 2, 7, 3, 8, 4.

Deze bewerkingen kunnen worden gecombineerd tot meer complexe manieren van schudden. Dat wordt op de volgende manier aangegeven:

1. De drie basisbewerkingen worden aangegeven met de letters b, r en l.
2. De bewerkingen staan genoteerd in de volgorde waarin ze moeten worden uitgevoerd.

3. Een geheel getal m ($1 < m < 10$) voorafgaande aan een bewerking betekent dat die bewerking m keer moet worden herhaald.
4. Een samengestelde bewerking kan tussen haakjes worden gezet met een getal m voor die haakjes, dat aangeeft dat die samengestelde bewerking m keer moet worden herhaald.

Voorbeelden:

- **lr** is een linker-rits, gevolgd door een rechter-rits.
- **4b** is een serie van vier keer een break.
- **4blr** is a serie van vier keer een break, gevolgd door een linker-rits en dan een rechter-rits.
- **4(blr)** betekent dat de serie “break, linker-rits, rechter-rits” viermaal wordt uitgevoerd.

Schrijf een programma dat twee regels leest van standard input. Op de eerste regel staat het getal N . Op de tweede regel staat een serie bewerkingen.

Je programma schrijft naar standard output één regel met daarop de volgorde van de kaartjes nadat de bewerkingen zijn uitgevoerd. De getallen worden telkens gescheiden door een spatie en ze geven de getallen op de geschudde stapel weer van boven af aan.

Voorbeelden

Invoer: 3 b	Invoer: 4 3br	Invoer: 4 3(br)
Uitvoer: 2 3 4 5 6 1	Uitvoer: 4 8 5 1 6 2 7 3	Uitvoer: 8 1 3 4 5 2 7 6

Randvoorwaarden:

Er zijn vijf groepen testgevallen; voor ieder testgeval dat je programma goed afhandelt krijg je 2 punten.

Testgevallen	N	Omschrijving	Punten
Reeks 1	$2 < N < 5$	Geen getallen in serie bewerkingen	4
Reeks 2	$2 < N < 5$	Geen haakjes in serie bewerkingen	6
Reeks 3	$2 < N < 20$	Geen haakjes in serie bewerkingen	8
Reeks 4	$2 < N < 20$	Geen beperkingen	10
Reeks 5	$2 < N < 50$	Geen beperkingen	12

De serie bewerkingen bestaat uit maximaal 60 tekens. Er staan nooit haakjes tussen haakjes.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Domino

B2. Meetpunt

B3. Woorden plaatsen

B4. Sterrenstelsels

Opgave C1. Nemo

De Nemo is een kleine onderzeeër die probeert niet gevonden te worden.

De oceaan wordt in deze opgave weergegeven als een rechthoek van R bij K cellen. Een cel wordt aangeduid met een '#' als er land is of met een '.' als er water is.

Voorbeeld (van een oceaan van 7 bij 9 cellen):

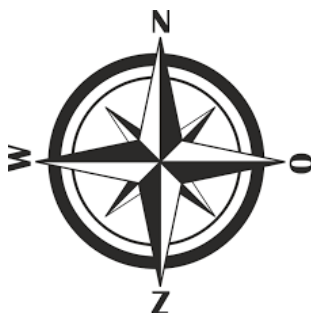
...##....

..#.##..#

..#....##

.##...#..

....#....



Elke minuut zendt de Nemo een radioboodschap uit waarmee hij aangeeft in welke richting de onderzeeër zich heeft bewogen. Die richting is altijd Noord (N), Oost (O), Zuid (Z) of West (W). en altijd is de afgelegde afstand per signaal één cel. Zie ook het plaatje voor deze richtingen.

Vera heeft een radar gemaakt waarmee ze deze periodieke signalen kan onderscheppen. Tijdens de laatste M minuten zijn er M radioboodschappen uitgezonden, die worden weergegeven als een string met M karakters, bijvoorbeeld 'WZ?00??'. Soms kan een signaal niet worden ontcijferd, dan staat er een '?'.

Vera weet de oorspronkelijke positie van de onderzeeër niet, maar wil gebruik maken van de kaart van de oceaan om de huidige positie uit te zoeken. Als je weet dat de Nemo elke minuut naar een aangrenzende cel beweegt, uitsluitend door het water reist en op de kaart blijft, moet jij Vera helpen om te bepalen op hoeveel verschillende posities de Nemo zich kan bevinden.

Schrijf een programma dat van standard input de volgende gegevens inleest.

- Op de eerste regels staan de getallen R, K en M, telkens gescheiden door een spatie.
- De volgende R regels bestaan elk uit K tekens '#' of '.', waarmee de oceaan wordt weergegeven.
- De laatste regel bestaat uit M tekens 'N', 'O', 'Z', 'W' of '?'; die geven de onderschepte radioboodschappen van de Nemo aan.

Je programma schrijft naar standard output een regel met daarop het aantal mogelijke posities dat de Nemo nog kan hebben.

Voorbeelden

Invoer:

5 9 7

...##.....

..#.#.#..#

..#.....##

.##...#..

.....#.....

WZ?00??

Uitvoer:

22

Invoer:

3 3 1

.#.

..#

#..

N

Uitvoer:

2

Invoer:

4 4 1

...#

...#

...#

#.#.

?

Uitvoer:

10

Randvoorwaarden

Testgevallen	Punten	Kenmerken van de testgevallen
Reeks 1	25	R, K en M zijn niet groter dan 100. Er zijn geen '?'-tekens in de invoer.
Reeks 2	40	R, K en M zijn niet groter dan 100. Er zijn ook '?'-tekens in de invoer.
Reeks 3	35	R en K zijn niet groter dan 500. Er zijn ook '?'-tekens in de invoer. M is niet groter dan 5000.

De tijdlimiet voor elk testgeval is 5 seconden.

Opgave C2. Bereik

Gegeven is een rechthoekig grid van R regels met elk K cellen; cellen in het grid bevatten getallen of zijn leeg. De opgave is sommige lege cellen zwart te kleuren, zodat precies aan de volgende voorwaarden is voldaan.

- Er zijn geen twee horizontaal of verticaal aangrenzende zwarte cellen.
- Van elke niet gekleurde cel kun je iedere andere niet gekleurde cel bereiken door telkens een stap opzij, omhoog of naar beneden te doen via uitsluitend niet gekleurde cellen.
- Voor elke cel met een getal erin geldt dat het getal aangeeft hoeveel ongekleurde cellen in een rechte lijn, dus horizontaal of verticaal, bereikbaar zijn vanaf die cel. De cel met het getal telt zelf een keer mee.

Voorbeeld van een grid van 9 regels met elk 6 cellen:

2					
			4		8
				7	11
5					9
8	10				
6		6			
					8

Oplossing:

2					
			4		8
				7	11
5					9
8	10				
6		6			
					8

Schrijf een programma dat van standard input inleest:

- Een regel met de getallen R en K, gescheiden door een spatie.
- R regels van elk K getallen, onderling gescheiden door een spatie. Een 0 betekent een ongekleurde cel zonder getal erin.

Schrijf naar standard output de oplossing in de vorm van R regels met elk K getallen, onderling gescheiden door spaties. Een 0 betekent een ongekleurde cel zonder getal erin, een -1 staat voor een zwarte cel.

Voorbeeld

Invoer:

```
9 6
2 0 0 0 0 0
0 0 0 4 0 8
0 0 0 0 0 0
0 0 0 0 7 11
5 0 0 0 0 9
8 10 0 0 0 0
0 0 0 0 0 0
6 0 6 0 0 0
0 0 0 0 0 8
```

Uitvoer:

```
2 -1 0 0 0 -1
0 0 -1 4 -1 8
-1 0 0 0 0 0
0 -1 0 0 7 11
5 0 0 -1 0 9
8 10 0 0 0 0
-1 0 -1 0 -1 0
6 0 6 0 0 0
-1 0 -1 0 -1 8
```

Randvoorwaarden:

Er geldt dat $8 < R < 17$ en $5 < K < 12$.

Voor de geplaatste getallen G in de cellen geldt $1 < G < R+K$.

Voor de helft van de testgevallen zijn R en K beide niet groter dan 10.

De tijdlimiet voor elk testgeval is 3 seconden.

Opgave D. Squares

Inleiding

Squares wordt gespeeld door twee spelers op een bord van 12x12 cellen waarop om beurten tegels van 2x2 cellen moeten worden geplaatst. Elke tegel bestaat uit vier cellen met elk een verschillende kleur. De kleuren worden aangegeven met de cijfers 1, 2, 3 en 4. Elke speler krijgt een eigen kleur toegewezen die geheim blijft voor de tegenstander; de tegenstander heeft een andere geheime kleur.

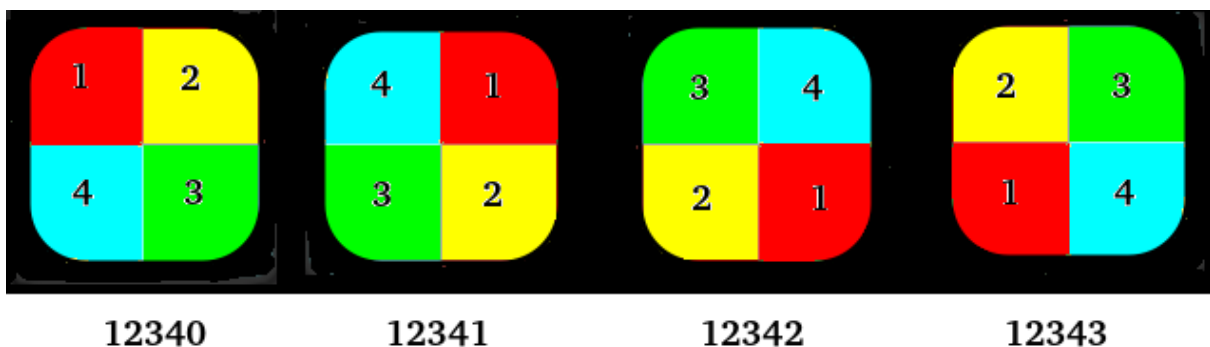
Het doel van het spel is om vierkanten met hoekpunten van je eigen kleur te vormen. Alleen vierkanten die evenwijdig zijn met de randen van het bord tellen mee. Grotere vierkanten leveren meer punten op (zie hieronder bij Punten). Het spel gaat door tot er geen tegels meer op het bord geplaatst kunnen worden. De speler met de meeste punten wint.

Het bord

De twaalf rijen worden aangeduid met de hoofdletters A tot en met L. De twaalf kolommen met de kleine letters a tot en met l.

De tegel

De tegels in dit spel zijn allemaal vierkant en hebben precies 4 verschillende kleuren in een specifieke volgorde. Je mag de tegels draaien. Dit wordt aangegeven met een cijfer 0..3, 0 betekent geen draaiing, 1 betekent 90 graden met de klok mee, etc. Hieronder staan alle 4 draaiingen van de tegel "1234" aangegeven:



Figuur 1. Alle draaiingen van de tegel "1234".

Als een speler aan de beurt is krijgt hij van de jurysoftware een gegenereerde tegel aangeboden met de vier cijfers 1,2,3,4 in willekeurige volgorde.

Die leest hij in van standard input.

Het eerste cijfer geeft de kleur linksboven aan voor de ongedraaide tegel, daarna ga je met de klok mee voor de andere kleuren. De speler mag zelf de draaiing van de tegel bepalen en doet dit door een cijfer 0, 1, 2 of 3 achter de tegel te plakken.

Het begin van het spel

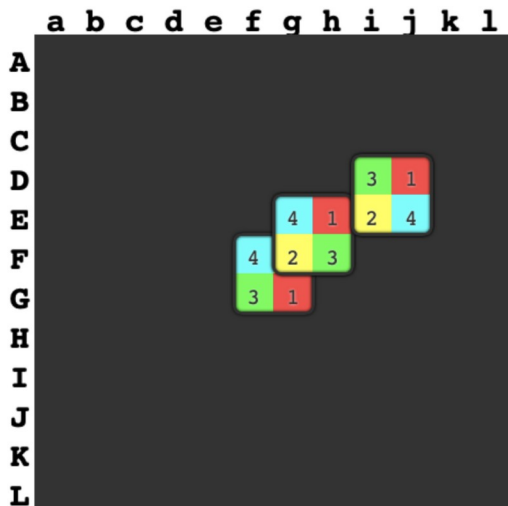
Beide spelers ontvangen eerst van standard input hun geheime kleur. Er wordt een starttegel gegenereerd, die in het midden van het bord wordt geplaatst. De cel linksboven van de starttegel komt dan op plek Ff. Zie figuur 2 tegel "42130". De volgorde van de kleuren in dit voorbeeld is 4123, deze cijfers worden geplaatst met de wijzers van de klok mee, te beginnen met de 4 op positie Ff. De laatste 0 geeft aan dat de tegel niet is gedraaid. Vervolgens lees je opnieuw een regel van standard input. Als op die regel het woord "Start" staat speel je als speler 1, anders bevat de regel de eerste zet van speler 1 en speel jij daarna als speler 2.

Een zet doen

De speler moet de tegel op het bord plaatsen. Er mag een overlap zijn met al geplaatste tegels van maximaal 1 cel. De kleur van de laatst geplaatste tegel geldt; het kan daarom zijn dat een al eerder gevormd vierkant verdwijnt omdat een hoekpunt een andere kleur krijgt. Als er geen overlap is, moet de tegel ergens worden aangesloten op de reeds geplaatste tegels. Minstens één van de cellen van de nieuw geplaatste tegel moet zich naast een andere gevulde cel van de groep tegels bevinden, horizontaal of verticaal.

De startspeler ontvangt na het woord "Start" een nieuw gegenereerde tegel. Het is aan de speler hoe hij deze op de bestaande tegel aansluit. Hij heeft in de beginsituatie in totaal 64 opties om dit te doen; ga dat zelf na.

Je programma heeft in totaal 30 seconden voor een spel. De tijd dat de tegenstander denkt telt daarbij niet mee.



In figuur 2 is de gegenereerde eerste tegel "4213" ongedraaid op Ff geplaatst. Speler 1 ontving tegel "4132" die hij ongedraaid op Eg heeft geplaatst. Dat geeft speler 1 aan door op standard output de regel "Eg0" uit te voeren. Speler 2 krijgt dit te lezen als "Eg41320". Vervolgens werd de tegel "1423" aan de tweede speler gegeven en plaatste hij deze als "Di14231". Zo gaat dat verder.

Figuur 2. Voorbeeld eerste zetten

Als een speler een foute zet doet, crasht, als het programma stopt voor het spel is afgelopen of teveel bedenktijd gebruikt (je programma heeft 30 seconden voor een spel), krijgt hij een "Quit" van de jurysoftware en neemt de jury zijn beurten over. De jurysoftware speelt alleen willekeurig gegenereerde zetten.

Als er geen tegels meer kunnen worden geplaatst is het spel afgelopen. Dan krijgt elke speler een "Quit" op zijn standaard input.

Voorbeeld communicatie bij figuur 2

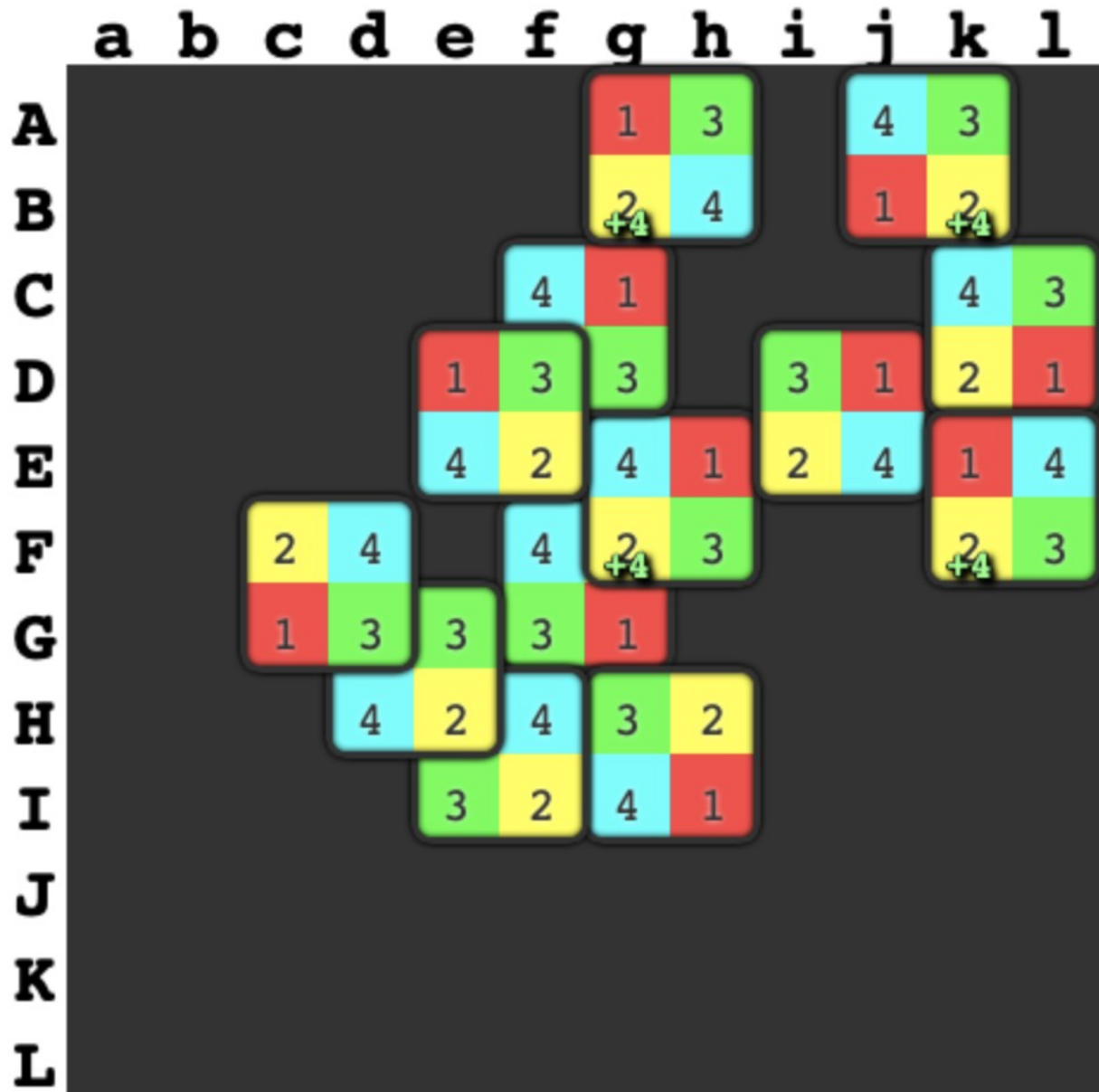
Speler 1		Speler 2		Toelichting
Invoer	Uitvoer	Invoer	Uitvoer	
1				Speelkleur speler 1
Ff42130				Starttegel
Start				Speler 1 begint
4132				Tegel voor speler 1
	Eg0			Zet van speler 1
		2		Speelkleur speler 2
		Ff42130		Starttegel
		Eg41320		Zet van speler 1
		1423		Tegel voor speler 2
			Di1	Zet van speler 2
Di14231				Zet van speler 2
3241				Tegel voor speler 1
	Cf2			Zet van speler 1
		Cf32412		Zet van speler 1
		3124		Tegel voor speler 2

Punten voor vierkanten

Elk vierkantje van je eigen kleur is aan het eind van het spel punten waard. Een vierkant van 2x2 is 1 punt waard, een vierkant van n x n is n - 1 punten waard. De maximale score voor een vierkant van 12x12 is 11 punten. Alleen vierkanten evenwijdig met de randen van het bord zijn punten waard. Je wint als je meer punten scoort dan je tegenstander.

Uitslag van het spel

De winnaar krijgt 200 punten plus het verschil in scorepunten. De verliezer krijgt 100 punten minus het verschil. Bij een gelijke stand ontvangen beide spelers 150 punten. Als het verschil groter is dan 100 punten, krijgt de verliezer geen punten. In dat geval krijgt de winnaar meer dan 300 punten.



Figuur 3. Speler 2 scoort 4 punten door het vierkant Bg-Bk-Fg-Fk te maken met geheime kleur geel (2).

In- en uitvoer

Je programma leest van standard input en schrijft naar standard output. Denk eraan dat je na iedere regel de uitvoer flusht. Zie ook de Technical Rules op nio.codecup.nl.

Als je programma als invoer de regel "Quit" krijgt moet je programma worden afgesloten.

Je programma mag informatie wegschrijven naar standard error. Je kunt als het spel gespeeld is na een testcompetitie gebruik maken van deze informatie.

Punten voor de olympiade

Als je programma samenwerkt met onze jurysoftware wordt het toegelaten tot het toernooi. In dat geval verdien je 20 punten voor deze opgave. Als je programma zonder fouten speelt kun je daarmee nog eens 50 punten verdienen.

De uitslag van dat toernooi is bepalend voor de laatste 30 punten; hoe hoger je programma eindigt, des te meer punten krijg je.

Inzenden van je programma gaat op nio.codecup.nl waar je eerst een account moet aanmaken.

Verantwoording:

Squares is een bordspel waarvan het principe bedacht is door Chris Handy. Zie: <https://www.perplext.com/packogame/box>.