



2004-2005 opgaven

Voorwaarden en aanwijzingen

De Nederlandse Informatica Olympiade is een wedstrijd voor leerlingen uit het voortgezet onderwijs. Aan de eerste en tweede ronde mag iedereen meedoen die dit schooljaar op het vmbo, havo of vwo zit. Aan de extra scholing en de derde ronde kunnen alleen die leerlingen meedoen die op 1 juli 2005 niet ouder zijn dan 20 jaar.

Het inzenden van je oplossingen

Als extra stimulans zijn er zes prijzen voor snelle inzenders van een correcte oplossing van opgave 1 (dat wil zeggen dat je met je programma tenminste 60 van de 100 punten haalt), één ter waarde van 500 euro en vijf van 100 euro. Om mee te dingen naar die prijs moet de oplossing van opgave 1 uiterlijk op 1 december 2004 binnen zijn. De uitslag volgt uiterlijk 10 januari 2005. De oplossingen van alle opgaven moeten op 10 januari 2005 binnen zijn. De uitslag volgt ongeveer een maand later. Deelnemers met een werkende inzending voor opgave 4 dingen mee naar de Windesheim Digitalisprijs: Een bedrag van 250 euro, beschikbaar gesteld voor de leerling of docent die in het toernooi bij deze opgave de beste plaats weet te bereiken.

Om deel te kunnen nemen aan de eerste ronde dien je je aan te melden op submit.informaticaolympiade.nl en daar bij je eerste bezoek je gegevens in te vullen. Je krijgt dan een gebruikersnaam en een wachtwoord. Als je op de site inlogt, kun je je oplossing inzenden via een webformulier. Voor opgaven 1, 2A, 2B, 3A en 3B wordt je programma binnen een week getest met een klein voorbeeld, waarmee je al vast 20 punten kunt verdienen; voor de rest van de punten wordt in januari 2005 een uitgebreider onderzoek gedaan door de jury, maar uitsluitend wanneer je programma die eerste test succesvol heeft doorstaan. Je inzending voor opgave 4 wordt getest, maar voor een definitief resultaat is het wachten op de wedstrijd om de CodeCup 2005, midden januari 2005.

De programma's:

- Gebruik een hogere programmeertaal, zoals Basic, C, C++, Java, Pascal, Perl of Python. Bij opgave 4 kun je alleen aan het bijbehorende toernooi meedoen als je een programma schrijft dat we door FreePascal 1.0.10, GCC 3.20 of Java J2SE 1.4.2 kunnen laten compileren. Als dat niet lukt, doet je programma wel mee aan de eerste ronde, maar niet aan het toernooi.
- Gebruik de bij de opgaven aangegeven programmanamen (bijvoorbeeld **ni01**), eventueel met een achtervoegsel dat aangeeft welke programmeertaal is gebruikt. Voor opgave 4 is geen naam voorgeschreven. Lever de programmatekst in, en zo mogelijk ook een gecompileerde versie (**.exe**-bestand of **.class**-bestanden in Java).
- Een beschrijving van de opzet van het programma mag als commentaar in de programmatekst worden opgenomen.
- Voor de opgaven 1, 2 en 3 geldt het volgende: Invoer staat altijd in een ASCII tekstbestand. Uitvoer moet altijd naar een ASCII tekstbestand. Gebruik altijd exact de bestandsnamen zoals vermeld in de opgave. Voor opgave 2C moet je een ASCII tekstbestand inleveren dat voldoet aan de specificaties van de opgave.
- Je programma mag geen bijzondere eisen stellen aan de hardware van de jury; het wordt getest op een Pentium III, 800 MHz met 256 MB intern geheugen en voldoende harde schijfruimte. Je programma zal worden gestart in een eigen directory, waar het lees- en schrijfrechten heeft. Voor opgave 4 gelden andere omstandigheden; zie daarvoor de tekst van de opgave en de site van de CodeCup.
- Je programma schrijft geen uitvoer naar het scherm.
- In Turbo Pascal 7.0 mag je de unit **crt** niet gebruiken.
- Bij iedere opgave staat een **tijdslimiet** aangegeven. Voor opgave 1, 2 en 3 geldt deze limiet op een PC met een Pentium-III met een 800 MHz processor en 256 MB intern geheugen. Oplossingen die te veel tijd vergen, krijgen geen punten. Overigens zijn deze limieten vooral om programma's te kunnen afbreken waarvan de jury niet verwacht dat ze nog een oplossing zullen vinden; we verwachten dat goede oplossingen altijd binnen de tijd gevonden kunnen worden.
- Bij iedere opgave mag je sourcecode maximaal 1,44 Mb groot zijn.
- Je programma mag niet meer dan 10 Mb geheugenruimte gebruiken.

Waardering en vervolg

Voor iedere opgave kun je maximaal 100 punten verdienen. Aan het eind van elke opgave staat een puntenverdeling aangegeven. Heb je minstens 200 van de 400 mogelijke punten gehaald, dan krijg je het Certificaat Nederlandse Informatica Olympiade uitgereikt en word je toegelaten tot de tweede ronde. De jury kan deze grens van 200 punten naar beneden bij stellen; daarom is het verstandig om toch in te zenden, ook als je denkt niet aan de 200 punten te komen.

Tijdens de tweede ronde zijn in ieder geval de volgende programmeertalen toegestaan: GCC 3.20, FreePascal 1.0.10 en JAVA J2SE 1.4. Over de wijze waarop je deze talen kunt gebruiken, komt informatie op de NIO-site te staan.

Vragen en inlichtingen

Op het World Wide Web is een site met allerlei informatie over de Nederlandse Informatica Olympiade. Hier staat achtergrondinformatie bij de opgaven. Ook kunnen hier aanvullingen, correcties en verduidelijkingen worden gepubliceerd. Daarnaast is het mogelijk je in te schrijven op de NIO-mailinglist. Het adres van de website van de Nederlandse Informatica Olympiade is: <http://www.informaticaolympiade.nl/>

Contactpersoon voor de eerste ronde is Willem van der Vegt, Christelijke Hogeschool Windesheim in Zwolle. Je kunt hem mailen naar willem@informaticaolympiade.nl

Opgave 1. Verborgen cijfers; controleren.

Deze puzzel is met toestemming ontleend aan het tijdschrift "Puzzelkampioen Logische Puzzels", een uitgave van Keesing Entertainmaint Media BV. Voor meer puzzels van deze uitgever zie www.denksport.nl

Verborgen cijfer-puzzels worden gemaakt op een diagram van 7 bij 7 vakjes. Probeer in elke rij en elke kolom de cijfers 1, 2, 3 en 4 te plaatsen. In elke rij en elke kolom komen de cijfers 1, 2, 3 en 4 elk één keer voor. Het cijfer aan de rand van de puzzel geeft het cijfer aan dat het eerste vanaf die zijde te zien is. Het kan best zijn dat er eerst één of meerdere lege vakjes komen! Een aantal vakjes is al ingevuld. Verder zijn er zwarte vakjes: hierop komt niets!

In opgave 3 word je uitgedaagd om deze puzzels met behulp van een computerprogramma op te lossen. In deze opgave moet je een programma schrijven om mogelijke oplossingen te controleren.

In de figuur hieronder zie je een voorbeeld van een goed ingevulde puzzel. Zie ook opgave 3 voor voorbeelden van nog oningevulde puzzels.

	1	3	1	2	3	4	2	
1	1	3				4	2	2
4		4	1	2	3			3
2	2	1	3	4				4
4	4	2				1	3	3
4			4	1	2	3		3
2			2	3	4		1	1
3	3				1	2	4	4
	3	2	2	3	1	2	4	

Opdracht:

Schrijf een programma **niol**. Invoer is een bestand **antwoord.in** dat bestaat uit negen regels tekst, elk van negen tekens. De invoer beschrijft een ingevulde puzzel van zeven bij zeven vakjes, met daar omheen een rand met de gegevens van de puzzel. Op de hoekpunten staat een 0. Ook lege, zwarte vakjes, worden weergegeven als een 0.

Je programma geeft als uitvoer een bestand **antwoord.uit** dat bestaat uit één regel. Daarop staat het getal 1 als de oplossing voldoet aan de omschrijving of het getal 0 als dat niet het geval is.

Voorbeelden:

Invoer (de cijfers langs de randen zijn vet afgebeeld)	Uitvoer	Toelichting
0 131234 20 11300042 2 4 0412300 3 22134000 4 4 4200013 3 4 0041230 3 20023401 1 3 3000124 4 0 3223124 0	1	De puzzel is goed ingevuld; op elke rij en kolom staan de cijfers 1,2,3 en 4 elk één keer en de informatie van de rand van de puzzel is goed gebruikt.
0 131234 20 11300042 2 4 0412300 2 22134000 4 4 4200013 3 4 0041230 3 20023410 1 3 3000124 4 0 3223124 0	0	Op de tweede regel van de puzzel moet als meest rechter cijfer een 2 zijn ingevuld! De zesde kolom van de puzzel bevat tweemaal een 1; in de zevende kolom staat geen enkele 1.

Punten voor opgave 1:

Bij inzending wordt je programma getest met twee voorbeelden. Deze voorbeelden worden ook gespiegeld en geroteerd aangeboden. Als je programma in alle gevallen het goede antwoord geeft, verdien je 20 punten.

Bij het testen in januari worden nog 8 andere testgevallen aan je programma voorgelegd. Voor elk testgeval dat in alle posities goed wordt beoordeeld kun je 10 punten verdienen.

Opgave 2. De kortste weg van A naar B.

Veel computergames spelen zich af op een zogenaamde grid-wereld: Het terrein, het speelveld, is een soort schaakbord, waarover de personages van vakje tot vakje bewegen. Eén van de problemen die gamedesigners moeten oplossen, is hoe personages zich efficiënt van het ene vakje naar het andere kunnen verplaatsen; in spellen waar dit niet goed is opgelost zie je de personages soms heel onlogische omwegen maken.

In deze opgave moet je een programma schrijven dat zo'n kortste weg op een speelveld kan vinden, als die er is. Het speelveld is een bord van zeven bij zeven vakjes; een deel van die vakjes kan worden gebruikt, een ander deel niet. Een weg bestaat uit wegdelen die twee aangrenzende vakjes met elkaar verbinden, horizontaal of verticaal. De lengte van een weg is het aantal wegdelen waaruit de weg bestaat (zie de voorbeelden hieronder). De afstand tussen twee vakjes is de lengte van de kortste weg tussen die vakjes.

De opgave bestaat uit drie onderdelen. Eerst moet je vanuit een gegeven beginpunt voor alle andere punten bekijken hoe lang de kortste weg is naar dat punt, als je het tenminste kunt bereiken. Vervolgens moet je in een gegeven diagram bepalen wat de grootste afstand is tussen twee punten. Tenslotte dagen we je uit een diagram te vinden waarin de afstand tussen twee vakjes zo groot mogelijk is.

Opdracht 2A:

Schrijf een programma **nio2a**. Invoer is een bestand **veld.in** dat bestaat uit zeven regels tekst, elk van zeven tekens. De invoer beschrijft een speelveld van zeven bij zeven vakjes. Bij de beschrijving worden 3 symbolen gebruikt:

- A beginpositie van de gevraagde wegen
- dit vakje mag voor de gevraagde wegen worden gebruikt
- X dit vakje is geblokkeerd en kan daarom niet voor de gevraagde wegen worden gebruikt

Uiteraard komt het symbool A precies één keer voor in de invoer.

Uitvoer is een bestand **veld.uit** van zeven regels. Daarin staat voor ieder vakje aangegeven wat de lengte van de kortste weg van A naar dit vakje is, als zo'n weg bestaat. Als er geen weg bestaat moet het getal -1 worden uitgevoerd. De weg van A naar A heeft lengte 0.

Alle getallen in de uitvoer worden op drie posities weergegeven, dus ze worden voorafgegaan door één of twee spaties.

De tijdlimiet voor deze opgave is 1 seconden.

Voorbeelden:

Speelveld

Invoer

```
-XX-----  
X-XX-  
-----X-  
-AX-XX-  
X-X-----  
X-X---X  
--XX--X
```

Uitvoer

```
-1 -1 -1 15 14 13 12  
-1 2 3 -1 -1 12 11  
2 1 2 3 4 -1 10  
1 0 -1 4 -1 -1 9  
-1 1 -1 5 6 7 8  
-1 2 -1 6 7 8 -1  
4 3 -1 -1 8 9 -1
```


```
X-----  
-XX-X--  
--AX---  
-X-X-X-  
--X-X-X  
---X---  
X-----X-
```

```
-1 -1 -1 -1 -1 -1 -1  
3 -1 -1 -1 -1 -1 -1  
2 1 0 -1 -1 -1 -1  
3 -1 1 -1 -1 -1 -1  
4 5 -1 -1 -1 13 -1  
5 6 7 -1 11 12 13  
-1 7 8 9 10 -1 14
```

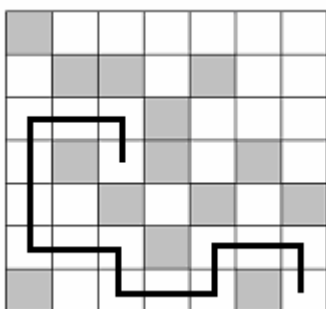
Opdracht 2B:

Schrijf een programma **nio2b**. Invoer is een bestand **leeg.in** dat bestaat uit zeven regels tekst, elk van zeven tekens. De invoer beschrijft een speelveld van zeven bij zeven vakjes. Bij de beschrijving worden 2 symbolen gebruikt; er is nu ten opzichte van opdracht 2A geen A in de invoer.

Je programma moet hierbij bepalen wat de grootste afstand is van twee vakjes die allebei niet geblokkeerd zijn. Uitvoer is een bestand **leeg.uit** van één regel. Op die regel staat een getal dat aangeeft hoe groot deze gevonden afstand weg is. Je hoeft dus niet te vertellen tussen welke vakjes deze weg te vinden is of hoe deze loopt, alleen de afstand is van belang.

Voorbeeld:

Speelveld



Invoer

```
X-----  
-XX-X--  
---X---  
-X-X-X-  
--X-X-X  
---X---  
X-----X-
```

Uitvoer

15

Opdracht 2C:

Maak een tekstbestand **best.in** met vrijwel dezelfde structuur als de invoer voor opdracht 2A. Nu wordt ook een symbool B gebruikt om de eindbestemming aan te duiden.

Het bestand moet voldoen aan de volgende voorwaarden:

- Het bestaat uit zeven regels met elk zeven toegestane tekens.
- Zowel de A als de B komen precies één keer in het bestand voor.
- Er moet een weg zijn van A naar B.
- De afstand van A en B moet zo groot mogelijk zijn.

Let op: Je moet geen programma insturen, maar alleen een bestand. Hoe je dat bestand hebt gemaakt doet er niet toe.

Als je een oplossing met de grootste afstand hebt gevonden, verdien je 20 punten voor de eerste ronde. Als je een geldige oplossing hebt gevonden die iets kleiner is, kun je ook nog punten verdienen (zie de puntenverdeling hieronder).

Punten voor opgave 2.

Als je het programma voor opdracht 2A inzendt, wordt het getest met 4 bestanden. Als het die test doorkomt heb je de eerste 20 punten binnen. Datzelfde geldt voor opgave 2B.

In januari worden er nog 6 testen met elk van beide programma's uitgevoerd. Daarmee kun je in totaal per programma nog 30 punten verdienen.

Voor opdracht 2C kun je maximaal 20 punten verdienen. Als jouw weg van A naar B lengte L heeft en de beste oplossing die wij kennen een lengte M heeft, wordt je puntenaantal berekend met de formule $75 * \frac{L}{M} - 55$. De uitkomst hiervan wordt afgerond op een geheel getal, en telt uiteraard alleen mee als je boven de 0 uitkomt.

Opgave 3. Verborgen cijfers. Nu zelf!

In opgave 1 heb je al kennis gemaakt met de “Verborgen cijfers”-puzzel.

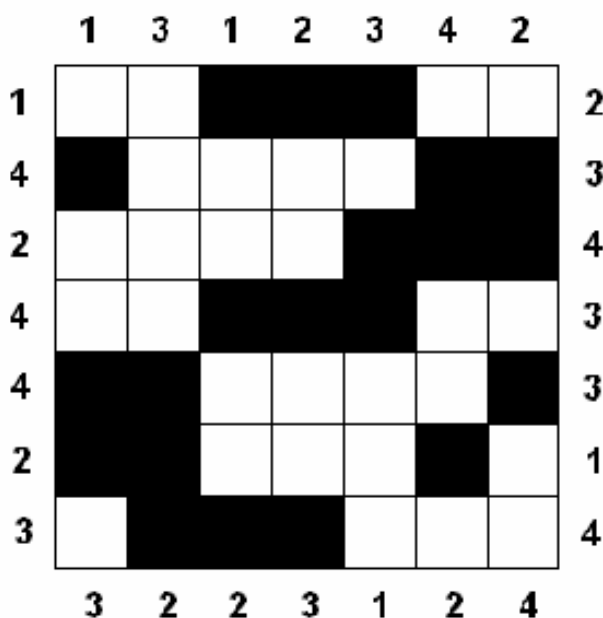
Een speciaal geval is het soort puzzel waarbij je alle zwarte vakjes al gegeven krijgt. Bij opdracht 3A moet je een programma schrijven dat dit eenvoudiger soort puzzels kan oplossen. Bij opdracht 3B kun je alle soorten puzzels aangeboden krijgen. Als je opdracht 3B tot een goed einde brengt moet dat programma ook in staat zijn de puzzels van opdracht 3A op te lossen. Je hoeft daarom alleen 3B in te sturen; als je dat niet lukt kun je 3A insturen voor eventueel 40 van de 60 punten.

Opdracht 3A en 3B:

Schrijf een programma **nio3a** of een programma **nio3b** dat een bestand **puzzel.in** inleest. Dit is een tekstbestand van 9 regels tekst van elk 9 tekens. De buitenste rand van het vierkant bestaat uit de getallen die aangeven welk cijfer je het eerst ziet wanneer je vanaf die kant kijkt in de betreffende rij of kolom; op de hoekpunten staat een 0. In het binnengebied staat een streepje als nog niet bekend is wat moet worden ingevuld, een cijfer als dat al bekend is en een 0 als al bekend is dat in een vakje niets mag worden ingevuld.

Een programma met de naam **nio3a** krijgt alleen invoer aangeboden waarbij alle zwarte vakjes al bekend zijn, zoals in voorbeeld 1. Een programma met de naam **nio3b** kan alle geldige puzzels als invoer aangeboden krijgen, dus ook die uit voorbeeld 2. Je programma zal uitsluitend worden getest met puzzels die tenminste één oplossing hebben!

Voorbeeld 1:



Bij deze puzzel hoort het volgende invoerbestand:

```
013123430
1--000--2
40----003
2----0004
4--000--3
400----03
200---0-1
3-000---4
032231240
```

Voorbeeld 2:

	1	3	1	2	3	4	2	
1								2
4		4						3
2				4				4
4							1	3
4				1				3
2								1
3								4
	3	2	2	3	1	2	4	

Bij deze puzzel hoort het volgende invoerbestand:

```
013123420
1-----2
4-4-----3
2---4-0-4
4--0--1-3
4---1---3
2-0---0-1
3-----4
032231230
```

De uitvoer is een bestand **puzzel.uit** dat bestaat uit zeven regels tekst van elk zeven tekens. Een leeg vakje wordt aangegeven met een 0, voor de ingevulde vakjes wordt het er geplaatste cijfer weergegeven (zie ook het voorbeeld). Als er meer dan één mogelijke oplossing is moet je er toch maar één geven.

De oplossing bij beide voorbeelden:

	1	3	1	2	3	4	2	
1	1	3				4	2	2
4		4	1	2	3			3
2	2	1	3	4				4
4	4	2				1	3	3
4			4	1	2	3		3
2			2	3	4		1	1
3	3				1	2	4	4
	3	2	2	3	1	2	4	

Tekstbestand met de oplossing:

```
1300042
0412300
2134000
4200013
0041230
0023401
3000124
```

De tijdlimiet bij deze opgave is 5 seconden.

Punten voor opgave 3.

Bij inzending wordt je programma met één invoerbestand getest; in dit bestand zijn de zwarte vakjes al gegeven. Als je programma daarbij de goede oplossing geeft, heb je de eerste 20 punten gehaald.

Als je die 20 punten hebt gehaald, worden in januari nog 4 andere testbestanden aan je programma als invoer gegeven. In één van deze vier testgevallen zijn alle zwarte vakjes al bekend.

Als je programma een goede oplossing vindt, levert dat per test weer 20 punten op.

Opgave 4. Lamistra.

Stratego is een spel van Hausemann & Hötte BV (Jumbo®) gevestigd te Amsterdam. Jumbo is auteursrechthebbend met betrekking tot het spel Stratego en alle bewerkingen daarvan. Jumbo is eigenaar van het merk Stratego voor spellen. Jumbo heeft toestemming gegeven voor de bewerking van Stratego door NIO in de vorm van Lamistra en de vervaardiging van programmatuur door de deelnemers in het kader van de NIO. Gebruik van Stratego en Lamistra anders dan in het kader van het maken van een programma en gebruik van het programma voor eigen gebruik, is niet toegestaan.

Waarschijnlijk ken je het spel Stratego: Twee spelers maken een geheime opstelling van hun stukken, en proberen vervolgens de vlag van de tegenstander te veroveren. Ministratego is een variant op een kleiner bord, met minder stukken, maar wel met dezelfde opzet. Latent Ministratego of Lamistra voegt er aan toe dat je de opstelling nog niet hoeft te hebben gekozen; pas als je slaat of wordt geslagen moet je de tegenstander duidelijk maken om welk type stuk het gaat. Natuurlijk moet de informatie die je dan geeft overeenstemmen met de geschiedenis van het spel tot dan toe.

Lamistra wordt gespeeld op een vierkant spelbord van 7 bij 7 vakjes. Deze vakjes hebben namen zoals in schaken; het vakje linksonder heet a1, het vakje rechtsboven g7. De vakjes b4 en f4 zijn zogenaamde meren, deze mag je niet betreden met je stukken.

Beide spelers hebben de beschikking over 21 stukken. Deze stukken krijgen tijdens het spel een rol. De mogelijke rollen zijn:

Rol	Aantal	Rang	Code	Bijzonderheden
Vlag	1	-	V	Het te veroveren stuk; mag niet verplaatst worden.
Bom	4	-	B	Mag niet verplaatst worden; kan alleen worden geslagen door de mineur.
Generaal	1	1	G	Hoogste officier; kan worden geslagen door de spion.
Overste	2	2	O	
Luitenant	3	3	L	
Mineur	4	4	M	Is in staat om bommen onschadelijk te maken.
Ruiter	5	5	R	
Spion	1	6	S	Is in staat om de generaal te slaan; verliest van alle andere stukken.

Rood begint met stenen op de vakjes a1 tot en met g3, blauw met stenen op a5 tot en met g7.

Een zet wordt aangeduid met een code van vijf tekens: De naam van het beginvakje, een minteken en dan de naam van het eindvakje (bijvoorbeeld a3-b3 of c2-c3).

Als je programma moet aangeven welke rol een stuk heeft gebruik je één hoofdletter uit het overzicht hierboven (zie ook het voorbeeld van in- en uitvoer).

a7	●	●	●	●	●	g7
●	●	●	●	●	●	●
●	●	●	●	●	●	●
a4	□	c4	d4	e4	□	g4
●	●	●	●	●	●	●
●	●	●	●	●	●	●
a1	●	●	●	●	●	g1

Voor Lamistra gelden de volgende regels:

- Spelers doen om beurten één zet; Rood begint.
- Een stuk beweegt van vakje naar vakje, één stap per zet, horizontaal of verticaal maar niet diagonaal.
- De meren op b4 en f4 zijn verboden gebied.
- Op ieder vakje kan maar één stuk staan.
- Een stuk mag niet in twee opeenvolgende zetten tussen twee vakjes heen en weer worden geschoven. Dit geldt niet als minstens één van beide zetten een aanval is.
- Als twee stukken van verschillende kleuren in aangrenzende vakjes staan, kunnen ze elkaar aanvallen. Dit is niet verplicht. De speler die aan de beurt is geeft aan dat hij wil aanvallen, hij duidt het aanvallende en het aangevallen stuk aan, en hij vertelt aan de wedstrijdleider welke rol zijn stuk heeft. Ook de tegenstander meldt welke rol het aangevallen stuk heeft.
- Als een stuk al eerder een rol heeft gekregen, moet deze gehandhaafd blijven.
- Als een stuk nog geen rol heeft gekregen, mag het uitsluitend een nog beschikbare rol krijgen.
- Een stuk dat zijn oorspronkelijke plaats heeft verlaten kan geen rol krijgen als vlag of bom; deze rollen horen bij stukken die niet kunnen bewegen!
- Het stuk met de laagste rang verliest de aanval en wordt van het bord verwijderd. Het winnende stuk wordt op de positie van het aangevallen stuk geplaatst.

- Als aanvaller en aangevallene dezelfde rang hebben, worden beiden van het bord verwijderd.
- Een generaal wint van een overste, een overste van een luitenant en dat gaat zo door; iedereen wint van de spion.
- Als de spion de generaal aanvalt wint de spion; als de generaal de spion aanvalt wint de generaal.
- Als een stuk een bom aanvalt verliest het de strijd. De uitzondering hierop is de mineur; als deze een bom aanvalt verliest de bom en neemt de mineur zijn plaats in.
- Een bom kan niet aanvallen, alleen maar aangevallen worden.
- Als van een speler de vlag wordt aangevallen heeft deze het spel verloren.
- Als een speler geen zet meer kan doen heeft hij het spel verloren.
- Een wedstrijd waarin beide spelers 200 zetten hebben gedaan zonder dat een winnaar bekend is wordt remise.

Opdracht:

Schrijf een programma dat invoer krijgt van `standard input` en uitvoer wegschrijft naar `standard output`. Je programma leest telkens één regel invoer, en geeft op basis van de dan bekende informatie een zet, en dat geheel wordt herhaald tot het spel is afgelopen; vervolgens moet je programma stoppen.

Als de eerste regel invoer het woord 'start' is, betekent dit dat je programma met de rode stenen speelt en als eerste een zet mag doen. Is de eerste invoer een zet, dan is dat de eerste zet van je tegenstander en speel je met de blauwe stenen.

Als je programma aan de beurt van zetten is, kies je een zet. Je geeft als uitvoer een regel met die zet. Als je zet een aanval is geef je een tweede regel met de rol van het aanvallende stuk, ook als deze rol al bekend is. Daarna stopt de tijd voor je programma, tot je tegenstander heeft gereageerd. Als je tegenstander een aanval heeft gedaan, geef je als uitvoer de rol van het aangevallen stuk, ook als deze al bekend is. Je krijgt daarna als invoer de rol van het aanvallende stuk. Vervolgens ben je aan de beurt van zetten.

Voor het spelen van een wedstrijd heeft je programma 180 seconden de tijd. Deze tijd wordt gemeten op een Pentium II, 200 MHz, 64 MB intern geheugen. Deze tijd mag naar eigen keuze over de verschillende zetten worden verdeeld. Alleen de tijd tussen het lezen van `standard input` en het schrijven naar `standard output` wordt geteld.

Je programma zal alleen geldige zetten aangeboden krijgen. Als jouw programma of het programma van de tegenstander een onregelmatigheid heeft begaan, krijg je als invoer een regel met de letter 'X' en moet je je programma afsluiten. Ook als je je vlag moet prijsgeven of de vlag van de tegenstander hebt geslagen, kan je programma worden afgesloten.

Voorbeeld:

Zet	Invoer Rood	Uitvoer Rood	Invoer Blauw	Uitvoer Blauw
Spelbegin	Start			
d3-d4		d3-d4		
			d3-d4	
d5-d4				d5-d4
				L
	d5-d4			
		R		
	L			
c3-d3		c3-d3		
			R	
			c3-d3	



110101001001010110110011101011010101101

CODECUP

1010110011101 **2004** 01001101101011

Parallel aan de Nederlandse Informatica Olympiade wordt een toernooi georganiseerd voor programma's die volgens bovenstaande specificaties het spel Lamistra kunnen spelen. Dit toernooi wordt ook wereldwijd aangekondigd onder de naam **CodeCup**. Er zijn prijzen voor de voorrondes en de finale, waaronder door Jumbo beschikbaar gestelde Strategospellen..

Het toernooi staat open voor alle belangstellenden, met uitzondering van de organisatoren van de Nederlandse Informatica Olympiade. De deelnemers aan de Nederlandse Informatica Olympiade doen zo veel mogelijk mee aan dit toernooi. Je moet je daarvoor zelf aanmelden op www.CodeCup.nl

Er wordt een competitie georganiseerd, die vrijwel live te volgen zal zijn op het Internet. De precieze competitievorm moet nog worden gekozen afhankelijk van het aantal deelnemers. Alle informatie over het toernooi kun je vinden op www.CodeCup.nl

Windesheim 

De Windesheim Digitalisprijs

De beste leerling of docent uit het Nederlandse voortgezet onderwijs in the **CodeCup** 2005 wint de jaarlijkse Windesheim Digitalisprijs van 250 euro. Deze prijs is beschikbaar gesteld door de afdeling contractactiviteiten VO/BVE van Windesheim Educatief.

Enkele wedstrijdregels

In iedere onderlinge wedstrijd zijn wedstrijdpunten te verdienen: Bij winst gaan er twaalf wedstrijdpunten naar de winnaar en geen punten naar de verliezer. Als beide spelers 200 zetten hebben gedaan zonder te winnen, worden de stukken op het bord geteld. Als één van beide spelers meer stenen heeft dan zijn tegenstander krijgt hij drie wedstrijdpunten en de tegenstander één; als beiden nog evenveel stukken op het bord hebben krijgen beide deelnemers twee punten.

Het doen van een ongeldige zet of het overschrijden van de bedenktijd betekent verlies van de partij; daarbij krijgt de verliezer een onreglementair partijeinde op zijn naam. De tegenstander heeft de partij automatisch gewonnen.

Winnaar van het toernooi wordt degene met de meeste wedstrijdpunten. Bij gelijk eindigen wint degene die de minste onregelmatige partijeindes heeft gekregen. Voor deelnemers die dan nog gelijk eindigen is de som van de door de tegenstanders behaalde wedstrijdpunten bepalend (de zogenaamde weerstandspunten). Als ook dat niet helpt, eindigen de deelnemers op dezelfde plaats.

Je programma wordt vooraf door de organisatie getest. Als het in staat is samen te werken met de wedstrijdsoftware, wordt het toegelaten tot het toernooi. Als dat niet het geval is, krijg je altijd bericht en wordt het programma apart gejureerd.

Het is mogelijk dat sommige wedstrijdregels nog moeten worden aangepast. Kijk regelmatig op www.codecup.nl voor de meest actuele versie van de regels!

Punten voor opgave 4.

Als je programma voor Lamistra wordt toegelaten tot het toernooi om de CodeCup, heb je daarmee de eerste 20 punten verdiend.

Tijdens het toernooi wordt bijgehouden of je ongeldige zetten doet; daarbij kan het gaan om tijdsoverschrijding en om zetten die niet kunnen (op het aangegeven vak staat geen stuk van jou, je slaat een stuk van jezelf, je zet buiten het bord, je neemt een duik in het meer, je beweegt een bom of de vlag, je beweegt meer dan 16 verschillende stukken, je benoemt een stuk dat al bekend is ineens anders of je schuift een stuk in twee opvolgende beurten heen en weer). Je kunt 60 punten verdienen door uitsluitend geldige zetten te spelen; voor elk ongeldig partijeinde verlies je 6 punten.

Tenslotte kun je maximaal 20 punten verdienen met de eindstand van het toernooi om de CodeCup. De beste NIO-deelnemer krijgt 20 punten, de volgenden krijgen steeds iets minder, maar als jij in het eindklassement andere NIO-deelnemers achter je hebt gelaten, verdien je in ieder geval punten voor dit onderdeel.

SPONSORS INFORMATICA OLYMPIADE

ONDER
NEDER
LANDSE
INFORMATICA
OLYMPIADE

PHILIPS

 **ELJAKIM**
Information Technology by



TU/e technische
universiteit
eindhoven

RuG


Universiteit Twente
de ondernemende universiteit

S | I | O

Windesheim 