

De eerste ronde Nederlandse Informatica Olympiade 2023-2024



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 2024 in Egypte.

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 2024 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 17 januari worden geüpload. Op 20 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 jurystelsel komen op 20 januari 2024 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 laatste jaarlijkse Windesheim Digitalisprijs van 200 euro. Inzenden mag tot 20 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. Ook is er een introductiecursus beschikbaar die is ontwikkeld voor mogelijke deelnemers aan de EGOI (European Girls Olympiad in Informatics): [Intro – C++ oefenen voor EGOI \(gitbook.io\)](https://gitbook.io)

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 17 januari 2024 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 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 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.

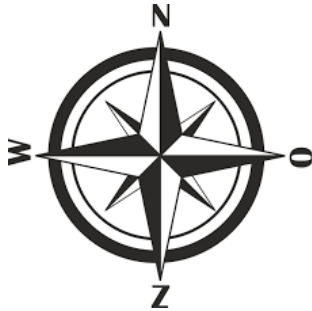
Opgave D en de CodeCup

Bij deze opgave moet je een programma schrijven dat het spel Sudoku kan spelen. Aan dit toernooi doen ook andere deelnemers me, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 20 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. De misplaatste robot

Een robotstofzuiger rijdt terwijl hij stofzuigt een route door het huis. De route is bepaald door een robotprogramma met K opdrachten. De robotstofzuiger zet zichzelf aan en voert het robotprogramma met K opdrachten uit. Daarna zet de robot zichzelf weer uit.

Er zijn vier mogelijke opdrachten, elk weergegeven met een hoofdletter N, Z, O of W, voor een beweging van één stap in de richting Noord, Zuid, Oost of West.



Het is de bedoeling dat de robot op dezelfde plaats eindigt als waar hij met zijn robotprogramma begonnen is. Helaas is er een aantal robotprogramma's meegeleverd waarbij dat niet het geval is.

Schrijf een programma dat van standard input twee regels inleest.

- op de eerste regel staat een getal K
 - op de tweede regel staat een robotprogramma van K opdrachten.
- K is een geheel getal met $0 < K < 256$.

Jouw programma schrijft naar standard output hoeveel opdrachten minstens moeten worden toegevoegd aan het robotprogramma om er voor te zorgen dat de robot aan het einde van zijn robotprogramma terug is op de plaats waar hij begon.

Voorbeelden

Invoer:
12
ZWNNOONZWWZ

Uitvoer:
2

Invoer:
7
NNNNNNN

Uitvoer:
7

Invoer:
4
NOZW

Uitvoer:
0

Voor je programma geldt een tijdslimiet van 1 seconde.

Opgave A2. Letterteller

Schrijf een programma dat één regel met tekst leest van standard input. Die tekst bestaat uit hoofdletters, andere tekens komen er niet in voor.

Het programma schrijft naar standard output één regel met daarin de letter of letters die het vaakst voorkomen in de invoer. Als er meerdere letters zijn die het vaakst voorkomen, dan geeft je programma die vaakst voorkomende letters op alfabetische volgorde.

Voorbeelden

Invoer:

LETTERTELLERTEKST

Uitvoer:

ET

Invoer:

AGBDEFC

Uitvoer:

ABCDEF

Invoer:

ALGEBRA

Uitvoer:

A

Randvoorwaarde:

De invoer bestaat uit N hoofdletters, met $1 < N < 256$.

Voor je programma geldt een tijdslimiet van 1 seconde.

Opgave A3. Talstelsels

Deze opgave gaat over de manieren waarop we positieve gehele getallen kunnen schrijven.

We schrijven getallen meestal in het decimale of tientallige positiestelsel. Een getal als 132 is opgebouwd uit eenheden (2 stuks), tientallen (3 stuks) en honderdtallen (1 stuks). Kortweg: $132 = 1 * 100 + 3 * 10 + 2 * 1$ of $132 = 1 * 10^2 + 3 * 10^1 + 2 * 10^0$. Het grondtal van het talstelsel is 10 en daarom gebruiken we ook 10 verschillende cijfers (0,1,2,3,4,5,6,7,8,9).

Als je een ander grondtal gebruikt is de opbouw van getallen vergelijkbaar. Om het onderscheid te maken schrijven we het grondtal als index achter het getal.

In het zestalig stelsel zijn 0,1,2,3,4,5 de zes cijfers die worden gebruikt. De index achter het getal is daarom $_6$ (een kleine 6). Na de 5_6 komt het getal 10_6 . In het tientallig stelsel wordt dit getal 10_6 genoteerd als $1 * 6^1 + 0 * 6^0 = 6_{10}$. Na 15_6 komt 20_6 , in het tientallig stelsel zijn dat 11_{10} en 12_{10} .

Het getal 132_6 in het zestalig stelsel kan in het tientallig stelsel worden omgeschreven door $1 * 6^2 + 3 * 6^1 + 2 * 6^0 = 56_{10}$ te berekenen.

In het twaalfalig stelsel heb je aan onze tien cijfers niet genoeg. Voor cijfers met een hogere waarde gaan we verder met de kleine letters uit het alfabet; a betekent dan 10_{10} , en b betekent 11_{10} .

Voor hogere grondtallen gaan we door tot en met de z. z_{36} heeft in het decimale stelsel een waarde van 35_{10} .

Het getal $aced_{15}$ is dus gelijk aan
 $a * 15^3 + c * 15^2 + f * 15^1 + d * 15^0 =$
 $10 * 3375 + 12 * 225 + 14 * 15 + 13 * 1 = 36.673_{10}$.

Voor deze opdracht is 36_{10} het hoogste grondtal. Ingewikkelder gaan we het niet maken.

Schrijf een programma dat drie regels leest van standard input.

- op de eerste regel staat het grondtal B (van bron) van het om te schrijven getal.
- op de tweede regel staat een getal G, geschreven in het talstelsel met grondtal B.
- op de derde regel staat het grondtal D (van doel) waarin getal G moet worden omgeschreven.

Het programma schrijft één regel uitvoer naar standard output. Daarop staat het getal G, maar nu omgeschreven in het talstelsel met grondtal D.

Voorbeelden:

Invoer:

6
132
10

Uitvoer:

56

Invoer:

10
132
17

Uitvoer:

7d

Invoer:

36
z
2

Uitvoer:

100011

Randvoorwaarde:

Er geldt $2_{10} \leq B \leq 36_{10}$, $2_{10} \leq D \leq 36_{10}$ en $0 < G < 1\ 000\ 000\ 000_{10}$

Voor je programma geldt een tijdslimiet van 1 seconde.

Bij de opgaven C1 en C2 wordt een ander getallensysteem verkend. Kennis van het positiestelsel kan je daarbij ook zeker helpen.

Opgave A4. De nacht van het nijlpaard

Nijlpaarden houden van een goede maaltijd. Maar ze hebben wel ruimte nodig voor zichzelf.

Elke nacht maakt de chef-kok N maaltijden klaar. Er is een lange tafel met aan één kant plaatsen die opend genummerd zijn. Elke maaltijd wordt willekeurig op één van die plaatsen neergezet.

De ceremoniemeester bepaalt daarna de minimale sociale afstand K tussen de nijlpaarden aan tafel. Als K bijvoorbeeld 2 is mogen twee nijlpaarden niet naast elkaar zitten maar moet er minstens 1 vrije plaats tussen hen in zitten .

De nijlpaarden komen binnen en kunnen gaan zitten op een plek waar een maaltijd staat als die plaats voldoet aan de minimale sociale afstand tot andere nijlpaarden. De ceremoniemeester zorgt ervoor dat het maximale aantal nijlpaarden deel kan nemen aan de maaltijd.

Schrijf een programma dat van standaard input

- eerst een getal N inleest ($1 < N < 100\ 000$);
- vervolgens een regel met het getal K inleest ($0 < K < 1\ 000\ 000$).
- daarna N regels met op iedere regel het plaatsnummer van een maaltijd inleest. Plaatsnummers zijn kleiner dan $1\ 000\ 000$.

Je programma geeft als uitvoer één regel naar standaard output met daarop het maximale aantal nijlpaarden dat deel kan nemen aan deze maaltijd.

Voorbeelden

Invoer:

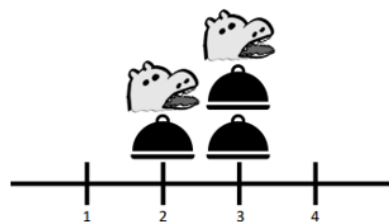
3
1
3
2
3

invoer grafisch:



Uitvoer:

2



Invoer:

3

2

1

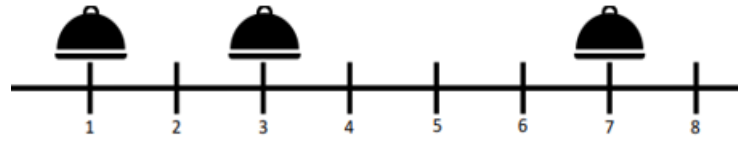
3

7

Uitvoer:

3

invoer grafisch:



Invoer:

5

4

3

5

1

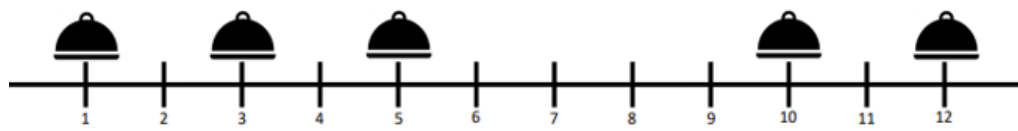
12

10

Uitvoer:

3

invoer grafisch:



Voor je programma geldt een tijdslimiet van 2 seconden.

Voor de testgevallen geldt het volgende:

3 testgevallen met $N = 2$

5 testgevallen met $K = 1$

9 testgevallen met $N < 1000$

3 testgevallen zonder beperkingen

Opgave A5. Lasersnijder

Cas heeft een programmeerbare lasersnijder waarmee hij vierkante platen triplex van N bij N centimeter gaat snijden.

Alle programma's beginnen in de linkerbovenhoek en volgen een serie van $2N$ instructies.

Die instructies zijn:

- beweeg 1 cm naar rechts, aangegeven met een 0
- beweeg 1 cm naar beneden, aangegeven met een 1.

Alle programma's eindigen in de rechterbenedenhoek.

Cas geeft de robot twee programma's op:

met programma A kan hij de linkerbenedenhoek en aansluitende delen verwijderen,

met programma B kan hij de rechterbovenhoek en aansluitende delen verwijderen.

In uitzonderlijke gevallen kan een programma ook niets afsnijden.

De programma's zijn zo ontworpen dat de snedes elkaar wel kunnen overlappen, maar elkaar nooit snijden.

Cas wil nu bepalen wat het grootste vierkant is dat hij uit het overgebleven hout kan maken. Het vierkant loopt evenwijdig met de zijden van het oorspronkelijke vierkant.

Schrijf een programma dat drie regels leest van standard input.

- op de eerste regel staat het getal N .
- op de tweede regel staat programma A van $2N$ tekens.
- op de derde regel staat programma B van $2N$ tekens.

Je programma schrijft naar standard output één regel met daarop de lengte van de zijde van het grootste vierkant dat uit het overgebleven hout kan worden gevormd.

Voorbeelden

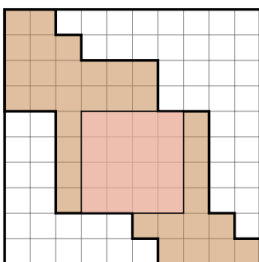
Invoer:

```
10
11110011110001010000
00101000110011110101
```

Uitvoer:

4

o

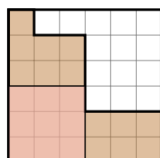


Invoer:

```
6
111111000000
0100111 00011
```

Uitvoer:

3

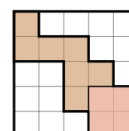


Invoer:

```
5
1100110100
0100101011
```

Uitvoer:

2



Randvoorwaarden:

In 90% van de testgevallen geldt $1 < N < 1000$. In de andere gevallen geldt $N < 50\,000$.

Voor je programma geldt een tijdlimiet van 2 seconden.

Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Op-en-neer-rij

B2. Muur

B3. Stapelen

B4. Kubus

Opgaven C: p-adische getallen

inleiding

De laatste tijd zijn p-adische getallen een interessant en belangrijk onderwerp op internet. Dit zijn gehele getallen die naar links toe oneindig ver doorlopen, twee voorbeelden hiervan zijn:

...33333 en ...66667

In dit voorbeeld zijn dit zijn getallen in het tientallig stelsel (zie voor uitleg over talstelsels opgave A3-Talstelsels). We kunnen deze 10-adische getallen (dat zijn dit soort getallen in het tientallig stelsel) schrijven als:

$\overline{3}_{10}$ en $\overline{67}_{10}$

Een streep boven een getal of groepje getallen betekent dat alles onder die streep oneindig vaak herhaald wordt.

Vermenigvuldigen

Deze getallen hebben in ons gewone tientallig stelsel een betekenis. Het tweede getal kan bijvoorbeeld met het 10-adische getal $\overline{03}_{10}$ vermenigvuldigd worden. Merk op dat dit ons gewone getal 3 is.

$$\begin{array}{r} \dots 66667 \\ \dots \underline{00003} \times \\ \dots 00001 \\ \dots 0000 \cdot \\ \dots \underline{000} \cdot \cdot + \\ \dots 00001 \end{array}$$

Dit getal kan geschreven worden als $x = \overline{01}_{10}$ en is ons gewone getal 1.

Omdat $\overline{67}_{10} \times 3 = 1$ moet de conclusie getrokken worden dat $\overline{67}_{10}$ de betekenis heeft van $1/3$. Een breuk dus.

Optellen en aftrekken

We kunnen onze twee voorbeeldgetallen ook bij elkaar optellen:

$$\begin{array}{r} \dots 66667 \\ \dots \underline{33333} + \\ \dots 00000 \end{array}$$

We krijgen het getal $\overline{0}_{10}$ oftewel het ons gewone getal 0 terug. Nu moeten we de conclusie trekken dat het 10-adische getal $\overline{3}_{10}$ de betekenis moet hebben van $-1/3$ omdat opgeteld bij $1/3$ er 0 uitkomt. 10-adische getallen kunnen dus gehele getallen zijn, maar ook breuken en zelfs negatieve breuken. Het getal -1 kunnen we vinden door 1 van 0 af te trekken:

$$\begin{array}{r} \dots 00000 \\ \underline{\dots 00001} - \\ \dots 99999 \end{array}$$

Het 10-adische getal $\overline{9}_{10}$ is dus ons gewone getal -1 .

Voorbeelden in andere stelsels dan het 10-talig stelsel

Nu blijkt dat als we niet in het 10-talig stelsel werken, maar in een stelsel met als grondgetal een getal p , dat dan die hele manier van rekenen ook blijkt te werken. Met dat in gedachten hebben we een paar opgaven voor je.

Het 2-adische getal $\overline{011}_2$ kan je lezen als:

$$\begin{aligned} & \dots + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = \\ & \dots 2048 + 512 + 128 + 32 + 8 + 2 + 1 \end{aligned}$$

Noem dit getal x . Vermenigvuldig x nu met $2^2 = 4$:

$$4x = \dots 8192 + 2048 + 512 + 128 + 32 + 8 + 4$$

Als je hier het getal x van aftrekt, dan gaan alle termen aan de voorkant van $4x$ en x tegen elkaar wegvallen:

$$3x = 4x - x = \dots (128 - 128) + (32 - 32) + (8 - 8) + 4 - 2 - 1 = 1$$

Daaruit volgt $3x = 1$ en x is dus de breuk $1/3$. Oftewel $\overline{011}_2 = 1/3$

Als voorbeeld, in andere stelsels kan je $1/3$ als volgt schrijven:

$$1/3 = \overline{23}_4 = \overline{132}_5 = \overline{45}_7 = \overline{374}_{11} = \overline{89}_{13} = \overline{5b6}_{17} = \overline{kl}_{31}$$

In dit voorbeeld zijn b , k en l de letters die de cijfers 11, 20 en 21 voorstellen bij gebrek aan voldoende cijfers (Zie ook opgave A3).

Dit kan je doen met vrijwel alle breuken, maar niet met allemaal! Zo bestaat de breuk $1/3$ niet als een 3-of 6-adisch geheel getal. Dit omdat de 3 van de noemer ook een deler is van grondtal 3 of 6. Een breuk is een p -adisch geheel getal als de noemer en het grondtal geen gemeenschappelijke delers hebben.

De opdrachten

We gaan je in opgave C1 vragen om een gewone breuk om te rekenen naar een p-adisch (geheel) getal. In opgave C2 vragen we je de omgekeerde berekening uit te voeren. We zullen je geen testgevallen voorschotelen die niet op te lossen zijn.

Op het internet zijn voldoende voorbeelden te vinden hoe je een gewoon getal kunt omrekenen naar een p-adisch getal en omgekeerd. Het is aan jou om uit te zoeken hoe je zo'n omrekening in een algoritme kunt omzetten en hoe je dat moet coderen.

Twee waarschuwingen:

- ChatGPT kan erg handig zijn, maar kan je ook volledig op het verkeerde been zetten door onjuiste informatie.
- Bij de omrekening van een p-adisch getal naar een gewoon getal heb je tijdens het coderen doorgaans hele grote integers nodig. Het blijkt dat voor de testgevallen in de laatste twee categorieën van opdracht C2 het gebruik van 64-bits integers niet voldoende is. Daar zal je een oplossing voor moeten bedenken.

De opgaven C1 en C2 doen een beroep op verschillende algoritmen en technieken. Voor beide opgaven kun je los van elkaar punten halen!

Opgave C1. Van breuk naar p-adisch getal

Schrijf een programma dat van standard input één regel inleest met daarop getallen G , T en N , telkens gescheiden door een spatie.

Het getal G stelt het grondtal van het talstelsel voor; $2 \leq G \leq 36$.

De getallen T en N zijn de teller en de noemer (in het gewone decimale talstelsel) van de breuk die omgerekend moet worden naar een G -adisch geheel getal. De grootste gemeenschappelijk deler van T en N zal gelijk zijn aan 1 en alleen de teller T kan negatief zijn; N en T zijn geheel en N is groter dan 0. Teller en noemer zijn een 64-bits integer.

Schrijf naar standard output de representatie van het G -adische gehele getal in het format $x|v$. De substring x stelt het repeterende deel voor van het G -adische getal en de substring v het vaste rechtergedeelte ervan. Als er geen substring v is, dan ontbreekt het gedeelte $|v$. Schrijf dan alleen de substring x als uitvoer.

Voorbeelden

Invoer:
10 1 3

Invoer:
5 -3 13

Invoer:
31 -3393 7

Uitvoer:
6|7

Uitvoer:
1034

Uitvoer:
qhm4d8|at

Toelichting

De breuk $1/3$ wordt geschreven als $\overline{67}_{10}$.

De breuk $-3/13$ wordt geschreven als $\overline{1034}_5$.

De breuk $-3393/7$ wordt geschreven als $\overline{qhm4d8at}_{31}$.

Randvoorwaarden

Testgevallen	Punten	Kenmerken van de testgevallen
Reeks 1	10	De testgevallen hebben het grondtal 10. r bestaat uit maximaal 6 karakters. v bestaat uit maximaal 1 karakter.
Reeks 2	20	De testgevallen hebben het grondtal 10. r bestaat uit maximaal 100 karakters. v bestaat uit maximaal 5 karakters.
Reeks 3	30	De testgevallen hebben een grondtal anders dan 10. r bestaat uit maximaal 6 karakters. v bestaat uit maximaal 1 karakter.
Reeks 4	40	De testgevallen hebben een grondtal anders dan 10. r bestaat uit maximaal 100 karakters. v bestaat uit maximaal 5 karakters.

De cijfers zijn genummerd in volgorde van 0 t/m 9 en daarna van a t/m z.

De tijdlimiet voor elk testgeval is 1 seconde.

Opgave C2. Van p-adisch getal naar breuk

Schrijf een programma dat van standard input een string X inleest.

De string X stelt een G -adisch getal voor in het format $g\ r|v$. Het getal g is het grondtal van het talsysteem. Daarna volgt een spatie. De substring r stelt het repeterende deel voor van het G -adische getal en de substring v het vaste rechtergedeelte ervan. Als er geen substring v is, dan ontbreekt het gedeelte $|v$.

Je programma schrijft naar standard output eerst een regel met een getal T weg en dan een regel met een getal N . T stelt de teller voor van de breuk en N de noemer. De grootste gemeenschappelijk deler van T en N moet gelijk zijn aan 1 en alleen de teller T mag negatief zijn. N is ook geen 0. De uitvoer is geschreven in het tientallig stelsel. De getallen T en N hebben maximaal 100 cijfers.

Voorbeelden

Invoer:

10 6|7

Uitvoer:

1

3

Invoer:

5 1034

Uitvoer:

-3

13

Invoer:

31 qhm4d8|at

Uitvoer:

-3393

7

Toelichting

Deze getallen komen overeen met de voorbeelden bij opgave C1.

Randvoorwaarden

Testgevallen	Punten	Kenmerken van de testgevallen
Reeks 1	10	De testgevallen hebben het grondtal 10. Berekeningen kunnen worden uitgevoerd met 64-bits integers. r bestaat uit maximaal 6 karakters. v bestaat uit maximaal 1 karakter.
Reeks 2	20	De testgevallen hebben een grondtal anders dan 10. Berekeningen kunnen worden uitgevoerd met 64-bits integers. r bestaat uit maximaal 6 karakters. v bestaat uit maximaal 1 karakter.
Reeks 3	30	De testgevallen hebben het grondtal 10. Berekeningen kunnen niet worden uitgevoerd met uitsluitend 64-bits integers. r bestaat uit maximaal 100 karakters. v bestaat uit maximaal 5 karakters.
Reeks 4	40	De testgevallen hebben een grondtal anders dan 10. Berekeningen kunnen niet worden uitgevoerd met uitsluitend 64-bits integers. r bestaat uit maximaal 100 karakters. v bestaat uit maximaal 5 karakters.

De cijfers zijn genummerd in volgorde van 0 t/m 9 en daarna van a t/m z.

De tijdlimiet voor elk testgeval is 1 seconde.

Opgave D. Sudoku

Inleiding

Dit jaar is de puzzel sudoku het spel van de CodeCup 2024. Daarbij hanteren we de gebruikelijke regels voor sudoku: een vierkant van 9 x 9 hokjes is opgedeeld in negen kleinere 3x3 vierkanten.

	a	b	c	d	e	f	g	h	i
A									
B									
C									
D									
E									
F									
G									
H									
I									

In iedere rij, in iedere kolom en in ieder 3x3-vierkant moeten cijfers 1,2,3,4,5,6,7,8,9 worden geplaatst zodanig dat er geen dubbelingen van dat cijfer in die rij, kolom of kleine vierkant voorkomen.

Het tweepersoons spel begint met een leeg bord met 9x9 hokjes. Om beurten moeten de spelers een geldig cijfer plaatsen in een lege cel. De eerste speler die denkt dat de sudoku een unieke oplossing heeft claimt de overwinning en wint het spel. Zie de paragraaf 'Een zet doen' voor verdere uitleg.

Doel van het spel

De uitdaging van dit spel is om als eerste vast te stellen dat de sudoku - waar beiden aan werken - een unieke oplossing heeft, voordat je tegenstander dat doet. Als je dat niet op tijd door hebt, is de kans groot dat je tegenstander dat zal claimen.

Niet elke voor de hand liggende zet mag worden gespeeld. Als de sudoku na een zet niet meer kan worden afgemaakt of opgelost, dan verliest de speler die die zet deed meteen. De speler die terecht claimt dat de sudoku nog precies één oplossing heeft wint; maar als er op dat moment nog twee of meer verschillende oplossingen mogelijk zijn, dan heeft die speler verloren.

Een zet doen

De rijen worden aangeduid met A tot en met I, de kolommen met a tot en met i. Een zet heeft het format "rij kolom cijfer". De zet "Dg7" plaatst bijvoorbeeld het cijfer 7 op vakje Dg.

Als je programma denkt dat de sudoku nog maar één oplossing heeft na jouw zet, kun je de overwinning claimen door een uitroepteken achter je zet te plaatsen, zoals "Hd7!".

Als je programma denkt dat de sudoku nog maar één oplossing heeft na de zet van je tegenstander is, kun je de overwinning claimen door alleen "!" als zet te plaatsen.

Score

Voor een spel mag je programma maximaal 30 seconden gebruiken, waarbij de tijd dat je tegenstander aan zet is niet mee telt.

Het is een hele uitdaging om een geldige zet te vinden. Het zou kunnen dat je programma te veel tijd nodig hebt omdat je lang bezig bent te controleren of de sudoku nog een oplossing heeft. Als je die controle achterwege laat kan het zijn dat je een ongeldige zet doet. In deze twee situaties zijn we vriendelijk voor je programma. De volgende uitslagen zijn mogelijk:

- Als je programma afsluit voor het einde van het spel, verlies je met 0-2.
- Als je programma de tijdlimiet overschrijft verlies je het spel met 1-2.
- Als je een onbegrijpelijke zet doet, een cijfer plaatst op een vakje dat al bezet is of een cijfer plaatst in een rij, kolom of vierkant waarin dat cijfer al is gebruikt, verlies je met 0-2.
- Als jouw zet de sudoku onoplosbaar maakt, verlies je met 1-2.
- Als je claimt dat de sudoku nog maar één oplossing heeft door een "!" te spelen, en het is inderdaad het geval, dan win je met 2-1. Als het niet klopt verlies je met 0-2.
- In het zeldzame geval dat geen van beide spelers de overwinning claimt is de eindscore een gelijkspel: 1-1.

Zodra er een uitslag wordt vastgesteld stopt het spel.

Als je een ongeldige zet doet die moeilijk was te controleren, of als je de tijdlimiet passeert, zullen we vriendelijk zijn en je alsnog 1 punt geven, Maar als je een ongeldige zet doet die je makkelijk had kunnen voorkomen zijn we niet zo aardig.

In- en uitvoer

Je programma leest van standard input en schrijft naar standard output. Als de eerste regel die je leest van standard input "Start" is moet je programma daarna de eerste zet uitvoeren. Als de eerste regel die je programma leest een geldige zet is, moet je de tweede zet doen en die uitvoeren.

Denk eraan dat je na iedere regel de uitvoer flusht. Zie ook de Technical Rules op www.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.

Voorbeeld

Eerste speler		Tweede speler	
invoer	uitvoer	invoer	uitvoer
Start	Ab3	Ab3	Gh6
Gh6	Ff2	Ff2	...
...
...	Ah8
Ah8	Fi5!	Quit	
Quit			

Ingezonden programma's nemen deel aan het toernooi om de CodeCup 2024. De beste leerling of docent uit het Nederlandse voortgezet onderwijs 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 zware fouten speelt kun je daarmee nog eens 50 punten verdienen; dat betekent dat je wel met 1-2 mag verliezen maar niet met 0-2. De uitslag van de competitie is bepalend voor de laatste 30 punten.

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

De jury controleert of je ook verder deelneemt aan de Nederlandse Informatica Olympiade en zal jouw status aanpassen aan NIO.

Tenslotte:

Er is op het moment van publicatie van deze opgaven nog een discussie over een mogelijke aanvulling op of uitbreiding van de spelregels. Houd daarom de website www.codecup.nl in de gaten!