



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 2023 in Hongarije.

## De eerste ronde

De eerste ronde van de Nederlandse Informatica Olympiade bestaat dit jaar uit 17 deelopgaven verdeeld in 3 series (A, B, C) en je score voor de CodeCup. Je hoeft niet alle deelopgaven te maken, maar dat mag natuurlijk wel!

### SCORES

Voor elke opgave die je inzendt en die wordt goedgekeurd kun je punten krijgen. De volgende tabel geeft aan hoeveel punten je maximaal per opgave kunt verdienen (bijvoorbeeld: als je opgave A3 goed hebt verdien je daarvoor maximaal 15 punten).

Opgave	A	B	C	CodeCup
1	5	10	10	
2	10	20	20	
3	15	30	30	
4	25	40	40	
5	25		100	
6	30			
7	40			
8	50			
<b>Totaal</b>	200	100	200	100

Je kunt daarnaast ook nog maximaal 100 punten verdienen door een inzending te doen voor de CodeCup 2023.

De punten voor de NIO hangen af van je ranking bij de CodeCup. Als je programma wordt geaccepteerd en bij de CodeCup minimaal één geldige wedstrijd speelt (los van de uitslag) krijg je in ieder geval 30 punten. De overige 70 punten hangen af van je ranking binnen de NIO deelnemers die deelnemen aan de CodeCup:

Resultaat CodeCup	Basis	Extra	Totaal
<b>Beste 10%</b>	30	70	100
<b>Beste 20%</b>	30	60	90
<b>Beste 30%</b>	30	50	80
<b>Beste 40%</b>	30	40	70
<b>Beste 50%</b>	30	30	60
<b>Beste 80%</b>	30	20	50
<b>Minimaal één geldige wedstrijd</b>	30	0	30
<b>Ongeldig programma</b>	0	0	0

### CERTIFICAAT

Je totale score kan dus maximaal 600 punten zijn. Afhankelijk van je score krijg je een certificaat met een vermelding erop.

Totaal score	vermelding
<b>500+</b>	goud
<b>350-499</b>	zilver
<b>200-349</b>	brons

De beste 100 leerlingen die minimaal 200 punten hebben behaald worden uitgenodigd om deel te nemen aan de tweede ronde in het voorjaar van 2023.

### DEELNEMEN

Om deel te kunnen nemen moet je een account maken op onze website:

<https://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 zorgvuldig om met de gegevens die je ons aanlevert. Wij zullen deze gegevens niet met derden delen. Indien je doorgaat naar de tweede ronde zullen we je naam (zonder verdere gegevens) wel doorgeven aan de universiteit voor de badges en deelname lijst.

Je kunt je uitwerkingen uploaden naar [submit.informaticaolympiade.nl](https://submit.informaticaolympiade.nl) wanneer je in het systeem bent ingelogd. 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 20 januari voor 22h00 worden geüpload. Op 21 januari wordt de eerste ronde gejureerd en kort daarna publiceren we de uitslag.

Voor de spelopgave, opgave D, moet je je aanmelden op [www.codecup.nl](http://www.codecup.nl) en kun je via die site ook je programma uploaden. De deelnemende programma's die meewerken met het jurysysteem komen op 21 januari 2022 tegen elkaar uit in een toernooi dat te volgen is op [www.codecup.nl](http://www.codecup.nl). Inzenden mag tot 21 januari 07h00. Dan begint het toernooi.

Voor alle opgaven geldt dat je ervan uit mag gaan dat je programma's alleen correcte invoer aangeboden krijgen.

### **OPGAVE A1 T/M A8**

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](http://cursus.informaticaolympiade.nl).

### **OPGAVE B1 T/M 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 het maximale aantal punten per opgave. Voor iedere dag later gaat er één punt van je score af. Inzendingen na 19 januari 2023 zullen niet worden verwerkt.

Als je een verkeerd antwoord hebt gegeven, verlies je meteen 20% van de maximale score, totdat er 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. Dat is niet altijd noodzakelijk. 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, kan oplossen.

### **OPGAVE C1 T/M C5**

Deze opgaven worden wat sneller moeilijk dan de A-opgaven.

### **OPGAVE D (CODECUP)**

Bij deze opgave moet je een programma schrijven dat het spel *Entropy* kan spelen. Aan dit toernooi doen ook andere deelnemers me, soms wel uit meer dan twintig verschillende landen. De programma's spelen op 21 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](http://www.codecup.nl)

In principe wordt je deelname door ons opgemerkt en wordt je aangemerkt als een NIO-deelnemer. Je kan ook even een e-mail sturen aan [contest@codecup.nl](mailto:contest@codecup.nl) en dan zorgen wij ervoor dat je een NIO- deelnemer wordt. Alle punten die je haalt in de CodeCup komen automatisch ook bij de puntentelling van [submit.informaticaolympiade.nl](http://submit.informaticaolympiade.nl) terecht.

### **EUROPEAN GIRLS' OLYMPIAD IN INFORMATICS (EGOI)**

Ben je een meisje en wil je kans maken om Nederland bij de European Girls' Olympiad in Informatics (EGOI) te vertegenwoordigen?

Geef je dan op via [girls@informaticaolympiade.nl](mailto:girls@informaticaolympiade.nl). We organiseren een programmeercursus als voorbereiding op de EGOI.

De cursus start op 1 oktober. Het is prima mogelijk om later in te stromen.

De beste vier deelnemers aan de tweede ronde van de NIO mogen deelnemen aan de EGOI in 2023 in Lund, Zweden.

### **INTERNATIONAL SCHOOL IN INFORMATICS JUNIOR (ISIJ)**

Bij voldoende interesse organiseren wij elk jaar een trip (met begeleiding) naar de International School in Informatics Junior. Deze scholing duurt tien dagen en is bedoeld voor leerlingen die 15-17 jaar oud zijn. De scholing wordt elk jaar in een ander land gehouden.

Als je interesse hebt: neem contact via [info@informaticaolympiade.nl](mailto:info@informaticaolympiade.nl); bij minimaal 5 deelnemers proberen wij deze scholing door te laten gaan.

### **WINDESHEIM DIGITALISPRIJS**

De leerling of docent uit het voortgezet onderwijs die het hoogste eindigt in het toernooi om de CodeCup 2023 ontvangt de Windesheim Digitalisprijs van 200 euro. Leerlingen die zich aangemeld hebben voor de eerste ronde van de NIO doen automatisch mee. Docenten die mee willen doen voor deze prijs moeten even een mail aan [contest@codecup.nl](mailto:contest@codecup.nl) sturen om aan te geven dat ze docent in het VO zijn.

# A-opgaven

Een palindroom is een woord dat hetzelfde blijft wanneer je de letters in omgekeerde volgorde afdruckt. Alle A-opgaven hebben iets met palindromen te maken.

Je kunt ervan uitgaan dat alle invoer nooit langer is dan 50 tekens. De invoer bestaat altijd uit lower-case letters uit het normale alfabet van 'a'..'z'. Er komen geen spaties voor in de invoer.

*Noot: voor alle A-opgaven geldt een tijdslimiet van 2 seconden*

## OPGAVE A1

Schrijf een programma dat van een regel van standard input een string leest. Het programma schrijft naar standard output een regel met daarop de omgekeerde string.

Voorbeeld

Invoer:       olympiade  
Uitvoer:       edaipmylo

## OPGAVE A2 - VAN LEP NAAR LEPEL

Schrijf een programma dat van een regel van standard input een string leest. Het programma schrijft naar standard output een regel met daarop het ingelezen woord, gevolgd door de spiegeling van het woord, waarbij je de laatste letter niet verdubbelt. Dit betekent in de praktijk dat je bij een woord van 1 letter geen letter toevoegt; en dus gewoon die ene letter teruggeeft.

Voorbeeld

Invoer:       lep  
Uitvoer:       lepel

Toelichting:

lep → pel → lep(p)el → lepel

## OPGAVE A3 - IS HET EEN PALINDROOM

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output '1' als de invoer een palindroom is en '0' als de invoer geen palindroom is.

Voorbeeld

Invoer:       parterretrap  
Uitvoer:       1  
  
Invoer:       stoeptegél  
Uitvoer:       0

#### **OPGAVE A4: LETTERS WEGHALEN**

Als je een woord hebt dat geen palindroom is dan kun je achteraan het woord letters weglaten om uiteindelijk op een palindroom uit te komen. Probeer zo min mogelijk letters weg te laten om dus een zo lang mogelijk palindroom over te houden.

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output een getal dat aangeeft hoeveel letters aan de achterzijde van het woord minimaal weggehaald moeten worden om een palindroom over te houden.

Voorbeeld

Invoer:        **parterretrap**

Uitvoer:       **0**

Invoer:        **lepels**

Uitvoer:       **1**

Invoer:        **auto**

Uitvoer:       **3**

Toelichting

In het eerste voorbeeld is er al sprake van een palindroom. Je hoeft dus 0 letters weg te halen. In het laatste voorbeeld kun je er echt geen palindroom van maken. Dat betekent dat je drie letters weg laat om maar 1 letter over te laten.

#### **OPGAVE A5 - LETTERS TOEVOEGEN**

Als je een woord hebt dat geen palindroom is dan kun je achteraan het woord letters toevoegen om uiteindelijk op een palindroom uit te komen. Probeer zo min mogelijk letters toe te voegen om toch een palindroom over te houden.

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output een getal dat aangeeft hoeveel letters aan de achterzijde van het woord minimaal toegevoegd moeten worden om een palindroom over te houden.

Voorbeeld

Invoer:        **parterretrap**

Uitvoer:       **0**

Invoer:        **lepe**

Uitvoer:       **1**

Invoer:        **auto**

Uitvoer:       **3**

Toelichting

In het eerste voorbeeld is er al sprake van een palindroom. Je hoeft dus 0 letters toe te voegen. In het laatste voorbeeld moet je in feite alle letters toevoegen, net zoals je ook deed in opgave A2. In het tweede voorbeeld hoef je maar één letter toe te voegen op lepel over te houden.

### **OPGAVE A6 - PALINDROMEN ZOEKEN**

Soms zitten er palindromen verstopt in een woord.

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output een getal dat aangeeft hoeveel palindromen er in de string verstopt zijn van lengte 5.

Voorbeeld

Invoer: parterretrap  
Uitvoer: 0

Invoer: lepel  
Uitvoer: 1

Invoer: abababa  
Uitvoer: 3

Toelichting

In het eerste voorbeeld zijn geen palindromen met lengte 5 te vinden. Het tweede woord is zelf een palindroom van lengte 5; dat is er dus exact één. In het laatste voorbeeld kun je drie palindromen vinden: 'ababa', 'babab' en 'ababa'.

### **OPGAVE A7 - UNIEKE PALINDROMEN ZOEKEN**

In opgave A6 ging je op zoek naar palindromen van lengte 5. In het laatste voorbeeld zaten drie palindromen, maar eigenlijk waren er twee exact hetzelfde. In deze opgave doe je hetzelfde als in de vorige opgave, maar je telt het aantal unieke palindromen.

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output een getal dat aangeeft hoeveel unieke palindromen er in de string verstopt zijn van lengte 5.

Voorbeeld

Invoer: parterretrap  
Uitvoer: 0

Invoer: lepel  
Uitvoer: 1

Invoer: abababa  
Uitvoer: 2

### OPGAVE A8 - ALLE PALINDROMEN ZOEKEN

Schrijf een programma dat van een regel van standaard input een string leest. Het programma schrijft naar standard output een getal  $n$  dat aangeeft hoeveel unieke palindromen er in de string te vinden zijn waarvan de lengte groter is dan 1. Schrijf vervolgens  $n$  regels met daarop de gevonden palindromen in lexicografische volgorde (dat betekent: de volgorde waarin de palindromen in een woordenboek zouden staan: 'a' komt voor 'aba', komt voor 'b'.)

Voorbeeld

Invoer: parterretrap

Uitvoer: 6  
arterretra  
erre  
parterretrap  
rr  
rterretr  
terret

Invoer: washandje

Uitvoer: 0

Invoer: zeep|lepel

Uitvoer: 3  
ee  
epe  
lepel



## B-opgaven

Je persoonlijke B-opgaven kun je zelf downloaden van [submit.informaticaolympiade.nl](http://submit.informaticaolympiade.nl).

Je krijgt een bestand dat bestaat uit de cijfers 0..9. Op de eerste regel staat hoeveel regels er volgen, en hoeveel cijfers er op elke regel staan. Je bent op zoek naar vierkanten waarbij op de hoekpunten van het vierkant hetzelfde cijfer staat. Voor elk cijfer moet je aangeven hoeveel vierkanten je hebt gevonden.

Lever een bestand in dat het totale aantal vierkanten bevat voor alle cijfers groter dan nul.

Bij opgave B1 geldt: je zoekt alleen orthogonale vierkanten. Dit betekent dat de zijden van het vierkant horizontaal en verticaal liggen.

Bij opgave B2 geldt: je zoekt alleen naar 'staande ruit'-vierkanten. Dit betekent dat de zijdes diagonaal lopen, en dat voor de hoekpunten geldt: de 'bovenste' ligt recht boven de 'onderste'; en de 'linkse' op dezelfde hoogte als de 'rechtse'.

Bij opgave B3 geldt: je telt zowel orthogonale als staande ruit vierkanten.

Bij opgave B4 geldt: elke oriëntatie mag; zolang de hoekpunten maar een vierkant vormen.

Als dit bijvoorbeeld het invoerbestand is:

```
7 8
0 0 0 0 0 0 0 0
0 1 0 1 1 0 0 0
0 0 2 1 1 1 0 0
0 1 0 1 1 3 2 0
2 0 0 3 2 0 0 0
0 0 0 0 0 0 3 2
0 0 2 0 3 2 0 0
```

Dan wordt hiermee de volgende rechthoek beschreven. (Voor het voorbeeld hebben we de '0' niet getoond, die doet bij de echte opgaven wel mee!).

	1		1	1			
		2	1	1	1		
	1		1	1	3	2	
2			3	2			
						3	2
		2		3	2		

Hieronder staat hoeveel vierkanten er in totaal gevonden worden voor de 1-2-3 cijfers. Bij de echte opgave moet je dus alle vierkanten van 0 t/m 9 tellen!

B1	B2	B3	B4
3	2	5	7

## C-opgaven

Een vriend van je is een tegelzetter. Hij heeft gevraagd of je hem wilt helpen bij wat nieuwe opdrachten. De lengte en breedte van een tegel is maximaal 1000. Als je meerdere tegels moet inlezen zijn dit er nooit meer dan 1000.

*Noot: voor alle C-opgaven geldt een tijdslimiet van 2 seconden.*

### OPGAVE C1

Je vriend wil van een individuele tegel graag weten wat de omtrek en de oppervlakte zijn.

Lees een regel in met daarop twee getallen, gescheiden door een spatie die de dimensies van de tegel aangeven.

Schrijf een regel met daarop twee getallen, gescheiden door een spatie. Het eerste getal geeft de omtrek van de tegel aan. Het tweede getal geeft de oppervlakte van de tegel aan.

Voorbeeld:

Invoer:        4 5  
Uitvoer:       18 20

Toelichting:

De omtrek van de tegel is  $4 + 4 + 5 + 5 = 18$  en de oppervlakte van de tegel is  $4 * 5 = 20$ .

### OPGAVE C2

Het is fijn dat je je vriend zo kon helpen. Maar hij komt toch best wel vaak met vragen over de omtrek en oppervlakte van de tegels. Schrijf daarom een programma dat voor een set van tegels bepaalt wat de omtrek en oppervlakte van elke individuele tegel is.

Lees eerst een regel in met daarop een getal  $n$  dat aangeeft hoeveel regels met tegels er volgen.

Op de volgende  $n$  regels staan telkens twee getallen, gescheiden door een spatie, die de dimensies van een tegel aangeven.

Schrijf  $n$  regels met daarop telkens twee getallen, gescheiden door een spatie. Elke regel beschrijft één tegel. Het eerste getal geeft de omtrek van de tegel aan. Het tweede getal geeft de oppervlakte van de tegel aan.

Voorbeeld:

Invoer:        3  
                  4 5  
                  2 10  
                  30 4  
Uitvoer:       18 20  
                  24 20  
                  68 120

### OPGAVE C3

Je vriend wil weten hoe ver hij kan komen als hij de tegels in een lange rij naast elkaar legt. Je mag de tegels 90 graden draaien. Wat is de maximale lengte die hij kan krijgen. En, bij deze lengte, wat is dan de breedte van het pad op het breedste en smalste punt?

Lees eerst een regel in met daarop een getal  $n$  dat aangeeft hoeveel regels met tegels er volgen.

Op de volgende  $n$  regels staan telkens twee getallen, gescheiden door een spatie, die de dimensies van een tegel aangeven.

Schrijf één regel met daarop drie getallen. Het eerste getal geeft aan wat de maximale lengte is van een pad als je de tegels tegen elkaar aanlegt. Het tweede getal geeft aan wat de breedte van dit pad is op het smalste punt. Het derde getal geeft aan wat de breedte van het pad is op het breedste punt.

Voorbeeld:

```
Invoer:      3
             4 5
             2 10
             30 4
Uitvoer:     45 2 4
```

### OPGAVE C4

Je vriend wil nu tegels op elkaar gaan stapelen. Maar hij wil nooit een grotere tegel op een kleinere tegel leggen. Specifiek betekent dit dat een tegel alleen op een andere tegel gelegd mag worden als zowel de lengte als de breedte van de bovenliggende tegel niet groter zijn dan die van de onderliggende tegel.

Als je op deze manier de tegels stapelt, wat is dan de hoogst mogelijke stapel die je kunt maken met de beschikbare tegels?

Lees eerst een regel in met daarop een getal  $n$  dat aangeeft hoeveel regels met tegels er volgen.

Op de volgende  $n$  regels staan telkens twee getallen, gescheiden door een spatie, die de dimensies van een tegel aangeven.

Schrijf één regel met daarop één getal  $h$ : het aantal tegels dat je maximaal op elkaar kunt stapelen.

Schrijf hierna  $h$  regels met telkens twee getallen. Elke regel beschrijft een tegel. Zorg ervoor dat je de tegel opschrijft in volgorde van beneden naar boven. De eerst genoemde tegel is dus de onderste (grootste) tegel. Zorg ervoor dat je de tegels alvast draait in de juiste richting. Dus voor elke regel geldt dat zowel de breedte als hoogte niet groter zijn dan op de vorige regel. Elke regel moet natuurlijk (al dan niet gedraaid) voorkomen in de lijst van de beschikbare tegels.

Als er meerdere oplossingen zijn hoef je er maar één te geven. In het bijzonder: voor de eerste tegel kun je natuurlijk altijd de oriëntatie kiezen. Het maakt niet uit welke oriëntatie je kiest, zolang alle bovenliggende tegels nooit groter zijn.

## OPGAVE C5

De stapeluitdaging heb je inmiddels opgelost. Je vriend wil nu de tegels nog steeds opstapelen, maar hij wil graag dat de stapels ongeveer even groot zijn zodat hij ze efficiënt kan opslaan in een garage.

De huisbaas van de garage geeft een boete afhankelijk van hoe de stapels zijn gemaakt. Als het aantal stapels  $s$  is, en de maximale hoogte van stapels is  $h_{max}$  en de minimale hoogte van deze stapels is  $h_{min}$ , dan is de boete  $s + h_{max} - h_{min}$ .

Kun je bepalen hoe je de tegels moet opstapelen om een zo laag mogelijke boete te krijgen? En wat is dan die boete?

Lees eerst een regel in met daarop een getal  $n$  dat aangeeft hoeveel regels met tegels er volgen.

Op de volgende  $n$  regels staan telkens twee getallen, gescheiden door een spatie, die de dimensies van een tegel aangeven.

Schrijf één regel met daarop drie getal:  $s, h_{min}, h_{max}$ . Hierna worden de  $s$  stapels beschreven. Dit doe je door telkens eerst een getal te geven dat aangeeft hoeveel tegels er op de stapel liggen. Op de volgende regels beschrijf je de tegels uit de stapel net als in opgave C4. Dus een regel per tegel, met als eerste de onderste (grootste) tegel van de stapel.

Voor deze opgave geldt dat  $n < 20$ . Je programma moet dan nog steeds binnen 2 seconden blijven om de volledige score te krijgen. Maar ook voor  $n < 10$  kun je al een deel van de punten verdienen.