

De eerste ronde Nederlandse Informatica Olympiade 2026-2027



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 2027 in Duitsland.

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 februari of maart 2027 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 21 januari worden geüpload. Op 23 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 23 januari 2027 tegen elkaar uit in een toernooi dat te volgen is op www.codecup.nl. Inzenden mag tot 23 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.

Voor de A-opgaven geldt een tijdlimiet van 1 seconde voor ieder programma.

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 21 januari 2027 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.

Voor de C-opgaven geldt een tijdslimiet van 3 seconden voor ieder programma.

Opgave D en de CodeCup

Bij deze opgave moet je een programma schrijven dat het spel Tumbleweed kan spelen. Dat is het spel van de CodeCup 2027. 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 spelen op 23 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. Piratenpraat

Inleiding

Iedereen weet dat piraten praten met veel letterherhalingen. We willen graag weten welke letter in een woord het vaakst voorkomt. In `arrrrrrghh` is dat de `r`, in `bibberrrig` zijn dat de `b` en de `r`.

Opgave

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

- Het woord is geschreven in kleine letters en bestaat uit minstens 3 en hoogstens 80 letters.
- Een woord bestaat uitsluitend uit letters zonder trema of accentteken. De `ij` telt als twee letters.
- Een woord bevat geen leestekens, spaties of andere toevoegingen.

Jouw programma schrijft naar standard output een regel met daarop de letter of letters die het vaakst voorkomen in de invoer. Als er meerdere letters even vaak voorkomen staan ze in de uitvoer op alfabetische volgorde.

Voorbeelden

Invoer: <code>arrrrrrghh</code>	Invoer: <code>bibberrrig</code>	Invoer: <code>lynx</code>	Invoer: <code>papegaaipoot</code>
Uitvoer: <code>r</code>	Uitvoer: <code>br</code>	Uitvoer: <code>lnxy</code>	Uitvoer: <code>ap</code>

Opgave A2. Controleletter

Inleiding

In Europa wil men graag een persoonsnummer voor alle inwoners invoeren. Zo'n nummer bestaat uit 10 cijfers en een controleletter.

Die letter wordt bepaald door de waarden van de cijfers en is er voor bedoeld dat er geen fouten worden gemaakt bij het gebruik van het nummer.

De controleletter kun je als volgt uitrekenen.

- Tel de cijfers op de oneven plaatsen bij elkaar op en vermenigvuldig het resultaat met 10. Noem dat getal G_o .
- Tel bij G_o de cijfers op de even plaatsen in de invoer op. Noem dat getal G_t .
- Bepaal de rest als je G_t deelt door 26. Noem die rest R .
- Tel 1 op bij R . Dat antwoord wordt C .
- De controleletter wordt nu de letter die op positie C staat in het alfabet (De A is 1, de B 2, de Z 26).

Opgave

Schrijf een programma dat een regel inleest van standard input. Op die regel staan 10 cijfers.

Je programma voert uit naar standard output één regel met daarop de controleletter. Die wordt geschreven als een hoofdletter.

Voorbeeld

Invoer

1274628033

Uitvoer

B

Toelichting

$$G_o = 10 \times (1 + 7 + 6 + 8 + 3) = 250$$

$$G_t = 250 + 2 + 4 + 2 + 0 + 3 = 261$$

$R=1$ (want 261 gedeeld door 26 is 10 met rest 1).

$C=2$

De tweede letter in het alfabet is de B.

Opgave A3. Wachtwoordcontrole

Inleiding

Veel websites en apps willen dat je tegenwoordig een sterk wachtwoord gebruikt. Jij wordt gevraagd om een programma te schrijven om kandidaat-wachtwoorden te controleren om aan te geven of ze sterk (genoeg) zijn en zo niet, waar het dan misgaat.

Een sterk wachtwoord voldoet aan de volgende eisen:

1. Het bevat tenminste een kleine letter
2. Het bevat tenminste een hoofdletter
3. Het bevat tenminste een cijfer
4. Het bevat tenminste een symbool, uit de collectie `+_) (*&^%$#@! . / , ; { }`
5. Het bestaat uit tenminste 12 tekens
6. Het bestaat uit ten hoogste 30 tekens

Het wachtwoord `"Aa1_"` (zonder de aanhalingstekens) voldoet wel aan de vier eerste eisen maar niet aan de vijfde.

Het wachtwoord `"Aa1234567890_"` voldoet aan alle eisen.

Opgave

Schrijf een programma dat een serie kandidaat-wachtwoorden controleert.

Op de eerste regel van de invoer staat een getal `N` dat aangeeft hoeveel wachtwoorden er volgen.

Daarna staat ieder kandidaat-wachtwoord op een regel.

Alle kandidaat-wachtwoorden bestaan uitsluitend uit hoofd- en kleine letters, cijfers en de toegestane symbolen.

De invoer bestaat uit maximaal 100 kandidaat-wachtwoorden.

Voor ieder kandidaat-wachtwoord wordt een regel uitgevoerd, met daarop het woord `"voldoet wel"` of de tekst `"voldoet niet"`.

In het geval van een ongeldig wachtwoord komt na het woord `niet` een spatie, gevolgd door het nummer van de eerste eis waaraan het kandidaat-wachtwoord niet voldoet.

Voorbeeld

Invoer

5
Aa1_
QWERTY1234567
Wateengezeur###
Wateengemis2##
LaatMeJeVergelijkenMetEenDagjeZomer&18

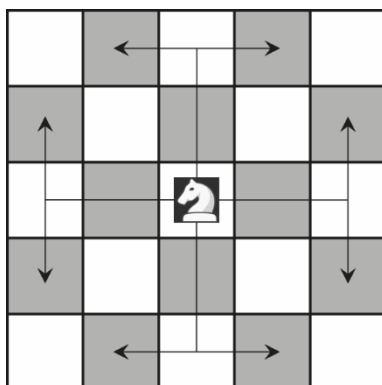
Uitvoer

voldoet niet 5
voldoet niet 1
voldoet niet 3
voldoet wel
voldoet niet 6

Opgave A4. Paardensprongen

Inleiding

In schaken is het paard het meest bijzondere speelstuk. Een zet met een paard bestaat uit twee stappen vooruit en één opzij. Zie de afbeelding hieronder.



Een schaakbord bestaat uit 8 x 8 velden. Elk veld heeft een naam, bestaande uit een letter en een cijfer. Hieronder zie je de namen van deze velden.

	a	b	c	d	e	f	g	h	
8	a8	b8	c8	d8	e8	f8	g8	h8	8
7	a7	b7	c7	d7	e7	f7	g7	h7	7
6	a6	b6	c6	d6	e6	f6	g6	h6	6
5	a5	b5	c5	d5	e5	f5	g5	h5	5
4	a4	b4	c4	d4	e4	f4	g4	h4	4
3	a3	b3	c3	d3	e3	f3	g3	h3	3
2	a2	b2	c2	d2	e2	f2	g2	h2	2
1	a1	b1	c1	d1	e1	f1	g1	h1	1
	a	b	c	d	e	f	g	h	

Het is niet altijd gemakkelijk te zien wat met een paard de kortste weg is tussen twee velden.

Opgave

Schrijf een programma dat twee regels leest van standard input.

Op de eerste regel staat de beginpositie A van een paard; op de tweede regel de gewenste eindpositie B.

Het programma schrijft één regel uitvoer naar standard output. Daarop staat aangegeven hoeveel zetten een paard minimaal nodig heeft om van A naar B te springen.

Voorbeelden

Invoer:

h1

a8

Uitvoer:

6

Invoer:

c4

c4

Uitvoer:

0

Invoer:

e1

e3

Uitvoer:

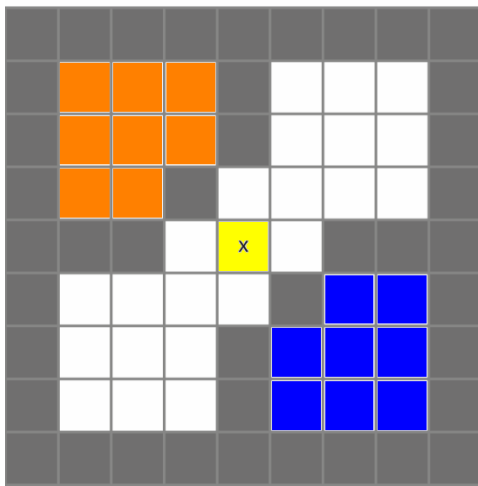
2

Opgave A5. Emmer

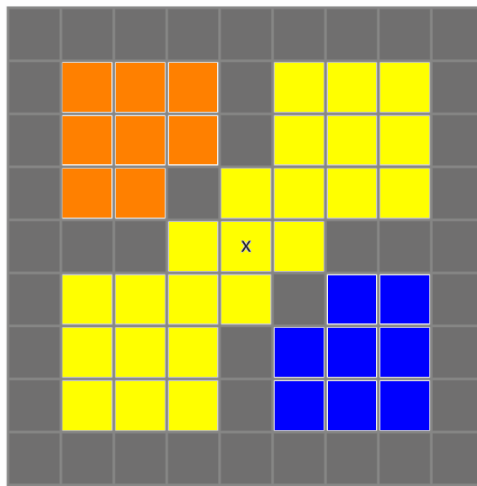
Inleiding

Een handige knop in een tekenprogramma is de verfemmer. Als je die actief hebt gemaakt en je klikt een pixel aan van de afbeelding, krijgt die pixel de actieve kleur, maar datzelfde geldt voor alle pixels die er mee verbonden zijn. Pixels zijn verbonden als ze horizontaal of verticaal naast elkaar liggen of als er een pad bestaat van verbonden pixels.

situatie 1



situatie 2



Een voorbeeld. Als je in het plaatje in situatie 1 vakje x een kleur geeft, worden alle vakjes in het aaneengesloten gebied gekleurd. De gebieden linksboven en rechtsonder gaan niet mee, die zijn alleen diagonaal met het gekleurde gebied verbonden.

Opgave

Schrijf een programma dat van standard input leest:

- Een regel met getallen R en K, gescheiden door een spatie; R is het aantal rijen van de afbeelding, K het aantal kolommen. R en K zijn allebei groter dan 0 en kleiner dan 1000.
- Vervolgens zijn er R rijen van elk K cijfers, die de kleuren aangeven van de pixels in de afbeelding. De gebruikte kleuren zijn aangegeven met 0, 1 tot en met 9.
- Tenslotte staan op een regel de getallen X, Y en T die aangeven dat de pixel in rij X (van boven naar beneden geteld, beginnend bij 1) en kolom Y (van links naar rechts gerekend, beginnend bij 1) kleur T moet krijgen (en het hele aaneengesloten gebied waar dat punt deel van uit maakt ook).

De uitvoer op standard output bestaat uit R rijen van K kolommen die beschrijven hoe de afbeelding er uit ziet na toepassing van de emmer-bewerking met de gegeven invoer.

Voorbeeld

Invoer

```
9 9
000000000
022201110
022201110
022011110
000111000
011110770
011107770
011107770
000000000
5 5 3
```

Uitvoer

```
000000000
022203330
022203330
022033330
000333000
033330770
033307770
033307770
000000000
```

Opgave B1 tot en met B4:

Download deze van submit.informaticaolympiade.nl

B1. Mozaïek

B2. Letterrij

B3. Ontcijferingen

B4. Parels rijen

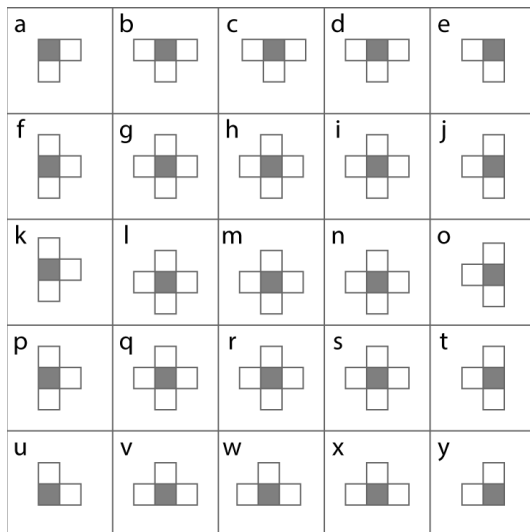
Opdrave C1. Klik

Inleiding

In een vierkant van 5x5 zijn 25 drukknoppen aangeduid met de letters a tot en met y.

Als je een knop indrukt veranderen de knop zelf en alle knoppen die er horizontaal of verticaal aan grenzen van kleur; zwart wordt wit of andersom. In afbeelding 1 staat dat in iedere knop schematisch weergegeven.

De gewenste eindtoestand is dat alle knoppen wit zijn, zoals in afbeelding 1.



Afbeelding 1. Eindtoestand

Opdrave

Schrijf een programma dat een regel inleest van standard input. Op die regel staan de knoppen die in de beginsituatie zwart zijn, op alfabetische volgorde.

Je programma schrijft op een regel van standard output welke knoppen er moeten worden ingedrukt om de gewenste eindtoestand (alle knoppen wit) te bereiken. Het aantal gebruikte knoppen moet daarbij zo klein mogelijk zijn.

Voor een goede oplossing waarbij je meer knoppen hebt gebruikt dan strikt nodig kun je nog steeds een deel van de punten krijgen.

Als de optimale oplossing K knoppen gebruikt en jij hebt een oplossing met M knoppen, waarbij M niet groter is dan 3*K, dan krijg je aan punten:

$$10^* \frac{(3M-K)^2}{(2K)^2}$$

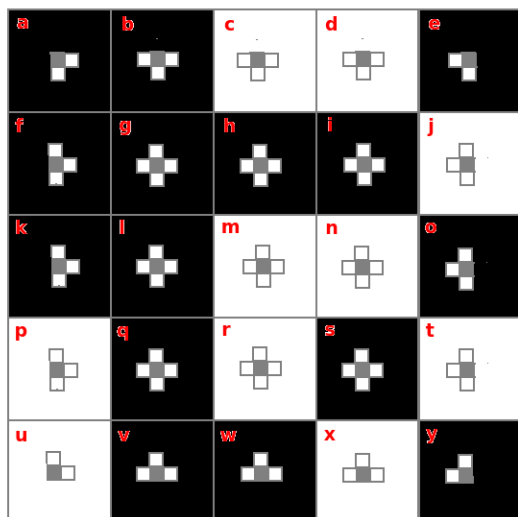
Voorbeeld

Invoer:

abefghikloqsvwy

Uitvoer:

aulqvmjot



Afbeelding 2. Voorbeeld

Als je een programma bij dit voorbeeld een oplossing van 13 keer op een knop drukken vindt verdien je in dit voorbeeld 6,05 punten. Een oplossing met 20 keer drukken geeft 1,51 punten.

Opgave C2. Kaarten ordenen

Inleiding

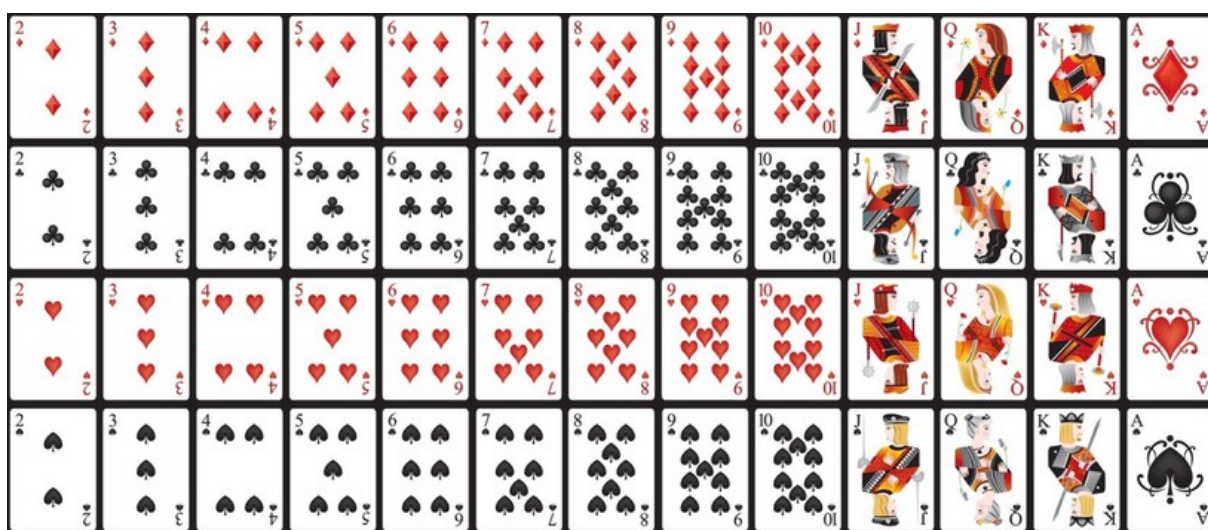
Bij het spel bridge krijg iedere speler 13 kaarten in een ongesorteerde stapel. Bij aanvang van het spel ordenen de spelers die kaarten om er beter overzicht over te krijgen.

Daarbij gelden de volgende uitgangspunten:

Kaarten van dezelfde speelsoort (schoppen, harten, ruiten of klaveren) worden meteen naast elkaar geplaatst.

De speelsoorten worden zo mogelijk afwisselend van kleur gegroepeerd (zwarte speelsoorten zijn schoppen en klaveren, rode soorten zijn harten en ruiten). Als er helemaal geen zwarte of helemaal geen rode kaarten zijn kan dit natuurlijk niet.

De kaarten van iedere speelsoort worden oplopend of aflopend van links naar rechts gesorteerd naar speelsterkte (Aas, Heer, Vrouw, Boer, 10 en dan aflopend tot en met 2). Die volgorde wordt voor alle speelsoorten gebruikt. Zie afbeelding 1 hieronder voor alle kaarten die per speelsoort op volgorde gesorteerd staan, van laag links tot hoog rechts.



Afbeelding 1. Alle speelkaarten per speelsoort

Om de 13 kaarten te ordenen voer je zogenaamde sorteerbewegingen uit. Dat betekent dat je een groepje kaarten dat naast elkaar ligt pakt en op een andere plaats terugzet.

Een sorteerbeweging wordt aangegeven met drie getallen:

- V Dat is de positie van de meest linker kaart van het groepje (de posities tellen van 1 tot en met 13)
- A Het aantal naast elkaar gelegen kaarten dat wordt gepakt en verplaatst
- N De nieuwe positie van de meest linkerkaart van het groepje na terugplaatsen

Opgave

Schrijf een programma dat van standard input een regel inleest met daarop 13 kaarten. Iedere kaart wordt weergegeven met twee tekens; eerst een S voor schoppen, H voor harten, R voor ruiten of K voor klaveren. Vervolgens een A voor een aas, een H voor een heer, een V voor een vrouw, een B voor een boer, een T voor een 10 of een 9, 8, 7, 6, 5, 4, 3 of 2. De invoer bepaalt de beginvolgorde van de kaarten. Tussen twee kaarten staat een spatie.

Je programma schrijft naar standard output eerst een regel met daarop het aantal sorteerbewegingen S dat nodig is om de invoer te ordenen. Daarna volgen S regels met de sorteerbewegingen die nodig zijn, telkens aangegeven met de getallen V, A en N, onderling gescheiden door een spatie.

Als je programma bij een testgeval het minimale aantal sorteerbewegingen M vindt krijg je voor die test het maximum van 10 punten. Als je programma meer bewegingen nodig heeft kun je nog een gedeelte van de punten voor het testgeval krijgen; voor ieder correct maar niet optimaal antwoord van S (met $M < S < 11$) sorteerbewegingen krijgt je programma $10 \cdot (0,6)^{S-M}$ punten.

Als na de sorteerbewegingen de kaarten niet gesorteerd zijn krijg je uiteraard geen punten.

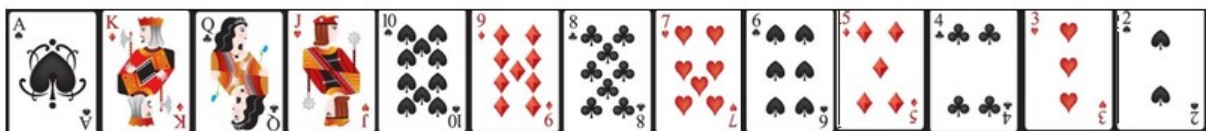
Voorbeeld

Invoer:

SA RH KV HB ST R9 K8 H7 S6 R5 K4 H3 S2

Uitvoer:

```
7
10 3 1
11 1 3
11 1 2
11 1 1
10 1 5
10 1 3
10 1 1
```



Afbeelding 2. Beginsituatie voorbeeld

In deze uitwerking worden de schoppen niet verplaatst. Eerst gaan drie kaarten tegelijk naar links, daarna worden de andere kaarten een voor een op een goede plek gezet.

In dit voorbeeld zijn er maar liefst 16 verschillende goede eindposities.

♠ A	♠ 10	♠ 6	♠ 2	♦ H	♦ 9	♦ 5	♣ V	♣ 8	♣ 4	♥ B	♥ 7	♥ 3
♠ A	♠ 10	♠ 6	♠ 2	♥ B	♥ 7	♥ 3	♣ V	♣ 8	♣ 4	♦ H	♦ 9	♦ 5
♥ B	♥ 7	♥ 3	♠ A	♠ 10	♠ 6	♠ 2	♦ H	♦ 9	♦ 5	♣ V	♣ 8	♣ 4
♥ B	♥ 7	♥ 3	♣ V	♣ 8	♣ 4	♦ H	♦ 9	♦ 5	♠ A	♠ 10	♠ 6	♠ 2
♦ H	♦ 9	♦ 5	♠ A	♠ 10	♠ 6	♠ 2	♥ B	♥ 7	♥ 3	♣ V	♣ 8	♣ 4
♦ H	♦ 9	♦ 5	♣ V	♣ 8	♣ 4	♥ B	♥ 7	♥ 3	♠ A	♠ 10	♠ 6	♠ 2
♣ V	♣ 8	♣ 4	♥ B	♥ 7	♥ 3	♠ A	♠ 10	♠ 6	♠ 2	♦ H	♦ 9	♦ 5
♣ V	♣ 8	♣ 4	♦ H	♦ 9	♦ 5	♠ A	♠ 10	♠ 6	♠ 2	♥ B	♥ 7	♥ 3
♠ 2	♠ 6	♠ 10	♠ A	♦ 5	♦ 9	♦ H	♣ 4	♣ 8	♣ V	♥ 3	♥ 7	♥ B
♠ 2	♠ 6	♠ 10	♠ A	♥ 3	♥ 7	♥ B	♣ 4	♣ 8	♣ V	♦ 5	♦ 9	♦ H
♥ 3	♥ 7	♥ B	♠ 2	♠ 6	♠ 10	♠ A	♦ 5	♦ 9	♦ H	♣ 4	♣ 8	♣ V
♥ 3	♥ 7	♥ B	♣ 4	♣ 8	♣ V	♦ 5	♦ 9	♦ H	♠ 2	♠ 6	♠ 10	♠ A
♦ 5	♦ 9	♦ H	♠ 2	♠ 6	♠ 10	♠ A	♥ 3	♥ 7	♥ B	♣ 4	♣ 8	♣ V
♦ 5	♦ 9	♦ H	♣ 4	♣ 8	♣ V	♥ 3	♥ 7	♥ B	♠ 2	♠ 6	♠ 10	♠ A
♣ 4	♣ 8	♣ V	♥ 3	♥ 7	♥ B	♠ 2	♠ 6	♠ 10	♠ A	♦ 5	♦ 9	♦ H
♣ 4	♣ 8	♣ V	♦ 5	♦ 9	♦ H	♠ 2	♠ 6	♠ 10	♠ A	♥ 3	♥ 7	♥ B

Afbeelding 3. De mogelijke correcte eindposities, elk op een regel.

Opgave D. Tumbleweed

Inleiding

Tumbleweed is een bordspel voor twee spelers, ontworpen in 2020 door Mike Zapawa. Sinds die tijd is er een bescheiden groep spelers, vaak afkomstig uit de Go-wereld, te vinden op [Discord](https://discord.com/invite/wu6Xdt497). Er zijn verschillende websites waarop je het spel kunt spelen tegen anderen of tegen een bot.

[1](https://tw.antontobi.com)

[2](https://abstractboardgames.com/tumbleweed/play)

[3](https://www.iggamecenter.com/en/)

[4](https://boardgamearena.com/gamepanel?game=tumbleweed)).

Er is ook een gids te vinden met wat tips over strategie:

[guide](https://guide.tumbleweed.games/).

Doel van het spel

Tumbleweed wordt gespeeld op een hexagonaal grid. De twee spelers worden aangeduid als Rood en Wit. Om beurten plaatsen de spelers een stapel van hun kleur in een cel met de bedoeling aan het einde van het spel zo veel mogelijk cellen te kunnen claimen. Een stapel kan een andere stapel zien als ze kunnen worden verbonden met een rechte lijn zonder dat er stapels tussen staan. Als een stapel wordt geplaatst wordt de waarde ervan bepaald door het aantal stapels van dezelfde kleur dat kan worden gezien. Nieuwe stapels moeten tenminste één stapel van dezelfde kleur kunnen zien. De speler kan een stapel van de tegenstander slaan (eigenlijk overnemen) of een eigen stapel opwaarderen als de nieuwe waarde strikt hoger is dan de huidige waarde van de stapel. Zetten waarbij een tegenstander een zojuist geslagen stapel onmiddellijk terugslaat zijn verboden.

Opstelling

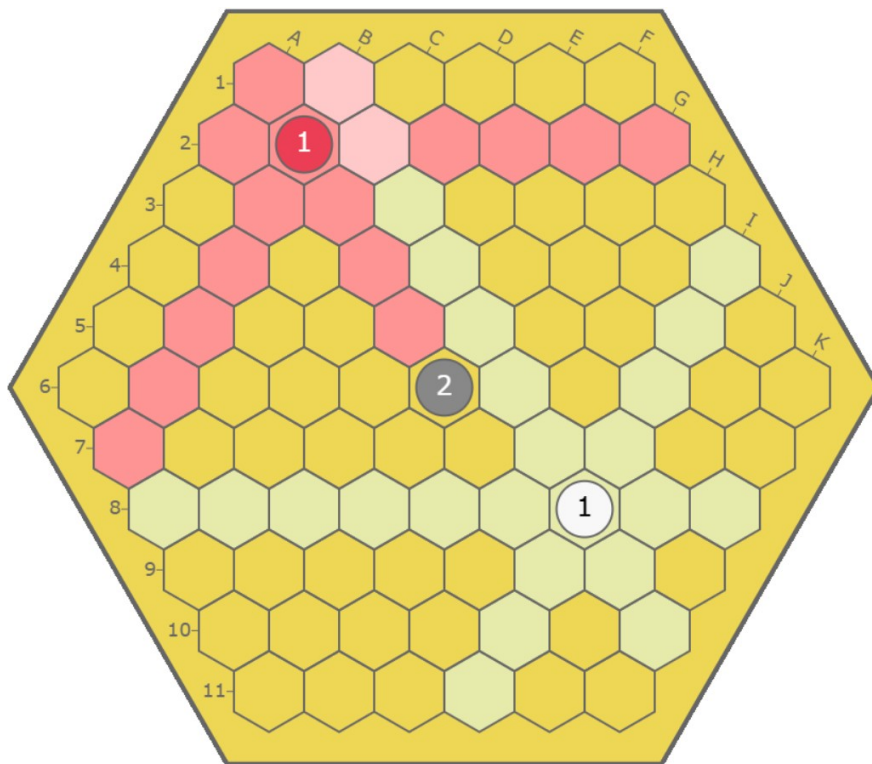
In de CodeCup spelen we op een hexagonaal grid met 6 zeshoeken aan iedere zijde. Een cel wordt aangeduid met een letter en een getal. De kolommen, die worden gerekend van noordoost naar zuidwest (/) zijn opvolgend A tot en met K, de rijen, die van west naar oost lopen zijn genummerd van 1 tot en met 11.

Bij het begin van het spel zijn er drie stapels op het bord:

- Een neutrale stapel met waarde 2 op het midden van het bord, in cel F6
- Een rode stapel met waarde 1
- Een witte stapel met waarde 1

De posities van de eerste rode en witte stapel worden bepaald door de jurysoftware aan het begin van het spel. In de competitie beginnen alle wedstrijden van dezelfde ronde met dezelfde beginopstelling zodat beide spelers de onderlinge wedstrijd met dezelfde positie kunnen spelen.

De jurysoftware zorgt ervoor dat de stapels in de beginopstelling niet aan de rand van het bord staan en niet pal naast elkaar.



Protocol

Je programma moet onderstaand protocol in acht nemen om met de jurysoftware samen te werken. Het programma moet informatie lezen van standard input en uitvoer schrijven naar standard output. Zie voor meer informatie de Technical rules (https://www.codecup.nl/rules_tech.php). Let erop dat je je uitvoer afsluit met een newline en daarna je output flusht.

Aan het begin van het spel leest de rode speler drie regels input.

Op de eerste regel staat de positie van de eerste rode stapel.

Op de tweede regel staat de positie van de eerste witte stapel.

Op de derde regel staat "Start" waarmee wordt aangegeven dat deze speler met rood speelt en mag beginnen. Daarna mag de speler de positie uitvoeren van de eerste stapel die hij zal plaatsen. De waarde wordt bepaald door de jurysoftware.

Daarna leest de witte speler drie regels input.

Op de eerste regel staat de positie van de eerste rode stapel.

Op de tweede regel staat de positie van de eerste witte stapel.

Op de derde regel staat de eerste zet van rood.

Vervolgens mag de speler de positie uitvoeren van de eerste stapel die hij zal plaatsen. De waarde wordt bepaald door de jurysoftware.

Vanaf dat moment geldt dat beide spelers om beurten de zet van de tegenstander lezen en zelf een zet uitvoeren. Als de invoer het woord "Quit" is in plaats van een gewone zet dan moet de speler zijn programma afsluiten.

De beginposities en de zetten worden aangegeven met twee delen: een hoofdletter voor de kolom en een getal voor de rij. De neutrale steen is bijvoorbeeld in positie F6.

Voorbeeld van de communicatie

Rood invoer	Rood uitvoer	Wit invoer	Wit uitvoer
B2		B2	
I8		I8	
Start	E5		
		E5	H7
H7	B4		
		B4	D3
D3	G4		
		G4	F7
...	...	...	...
Quit		Quit	

Een voorbeeld van een spel kun je zien op [CodeCup 2027 - Sample game](#)

Einde van het spel

Na iedere zet voert de jurysoftware twee simulaties uit om te zien hoeveel territorium iedere speler heeft.

Het territorium voor rood wordt gevonden door eerst alle mogelijke zetten voor wit na elkaar uit te voeren. Daarna worden de mogelijke zetten voor rood na elkaar uitgevoerd. De cellen met rode stapels erop na deze simulatie horen tot het rode territorium.

Op precies dezelfde manier wordt het territorium van wit bepaald; eerst alle zetten van rood, dan die van wit. Pas als iedere cel tot een territorium behoort beëindigt de jurysoftware het spel. Beide spelers krijgen het woord “Quit” te lezen en dienen hun programma af te sluiten. Op deze manier is het zeker dat een speler die aan de beurt is altijd nog een geldige zet kan doen.

Als een speler een ongeldige zet doet, bijvoorbeeld door een stapel te plaatsen op een cel met een ongeldige aanduiding of in een cel waar geen bevriende cel zichtbaar is, eindigt het spel. Beide spelers krijgen “Quit” te lezen. Als de speler de tijdlimiet van 30 seconden overschrijdt wordt zijn programma afgebroken; de tegenstander leest ook dan “Quit”.

Score

Elke speler krijgt 1 punt voor iedere cel in zijn territorium. De speler met het grootste territorium krijgt een bonus van 200 punten en de verliezer krijgt 100 punten. Een speler die het spel onreglementair afsluit krijgt geen punten. De tegenstander wordt dan automatische winnaar en krijgt 200 punten plus de punten voor zijn territorium; maar minimaal 246.

Visualisering

Bij het tonen van het spelverloop verstrekken we verschillende soorten informatie. Het hexagonale grid met de cellen en de coördinaten worden afgebeeld. In de cellen staan de geplaatste stapels met hun waarde weergegeven. De laatste zet is te herkennen aan een zwarte rand om de cel. De achtergrondkleur van een cel geeft aan wie er een stapel kan zetten, rood, wit of roze (dat geeft aan dat beide spelers een stapel kunnen plaatsen). Tegen het einde van het spel zal elke speler een territorium opbouwen. Dat wordt aangegeven door een rode of witte rand om het al opgebouwde territorium.

Onder het spelbord staat een scorebalk met verschillende segmenten. De gestreepte stukken geven aan hoeveel zetten maar een van beide spelers kan doen. Naar mate het spel vordert wordt ook aangegeven hoe groot het territorium van beide spelers is.

Zie ook [CodeCup 2027 - Rules of Tumbleweed](#) > Visualisation

Verschillen met de officiële spelregels

In de officiële regels wordt de beginpositie met een ander mechanisme bepaald. In de CodeCup bepaalt de jurysoftware de beginpositie.

De manier waarop het spel eindigt is ook anders geformuleerd; de simulaties waarmee de territoria worden bepaald zijn nodig om bij ieder spel een uitslag te genereren.

Tenslotte laten de officiële spelregels zogenaamde suicide-zetten toe, zetten die onmiddellijk weer door de tegenstander kunnen worden geslagen. Het is mogelijk aan te tonen dat dergelijke zetten je nooit vooruit kunnen helpen in het spel. Daarom zijn de regels voor het plaatsen van stapels aangepast om dergelijke zetten te voorkomen.

Voor de CodeCup-competitie

Je programma mag informatie wegschrijven naar standard error. Je kunt, als het spel gespeeld is, na een testcompetitie gebruik maken van deze informatie. De lengte van de stderr bestaat uit maximaal 10000 karakters.

Competitie

In de competitie tref je iedere tegenstander tweemaal, eenmaal als ROOD en eenmaal als WIT. De competitie wordt gewonnen door de speler die het hoogste aantal punten verzamelt tijdens de competitie. Meer informatie hierover is te vinden op www.codecup.nl in de Competition Rules.

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 www.codecup.nl waar je eerst een account moet aanmaken. Als je meedoet moet de CodeCup wordt dat door de organisatie opgemerkt en worden je punten meegenomen op de website submit.informaticaolympiade.nl.