

Opdracht 1. Een patiencespel

Het volgende patiencespel kun je spelen met behulp van een spel kaarten zonder jokers.

Je schudt de kaarten, en legt ze in het onderstaande patroon.

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

Doel van het spel is zo weinig mogelijk kaarten overhouden.

Je mag alleen kaarten weghalen die helemaal links of rechts in hun rij liggen.

Je mag altijd alleen maar twee kaarten tegelijk weghalen, waarvan de hoogte hetzelfde is (bijvoorbeeld twee boeren of twee vieren).

Het weghalen van twee kaarten noemen we een zet.

In de figuur hierboven zou je bijvoorbeeld ♠ 7 en ♥ 7 kunnen wegnemen.

Invoer:

Invoer bij alle opdrachten is een bestand `spe1.in` dat bestaat uit 8 regels. Daarop staan de kaarten van de 8 rijen aangegeven, met twee tekens per kaart: Een S, H, R of K voor schoppen, harten, ruiten of klaveren, en dan een A, H, V, B, T, 9, 8, 7, 6, 5, 4, 3 of 2. (De 10 wordt met een T aangeduid).

Voorbeeld bestand `spe10.in`:

```
S7S4R2
S5K8K3R6
S6STS2HVSV
HTH6H5H9H3R8
RVR9KTHHKVH2R5
RHRARBSHSAKBKHKKA
R4R7SBK7K4HBK6H8H7
S9R3RTK5K9S8K2HAH4S3
```

Dit bestand wordt bij de voorbeelden bij de opgaven als invoer gebruikt.

Uitvoer:

In uitvoerbestanden zul je soms zetten moeten aangeven. Dat doe je door de twee kaarten die je wilt weghalen achter elkaar op één regel te plaatsen, zonder spatie of scheidingstekens. Je begint altijd met de kaart op de hoogste rij van het diagram, en als het gaat om twee kaarten van dezelfde rij begin je met de kaart aan de linkerkant.

Voorbeeld van een zet: `S7H7`

Voorbeeldbestanden en testen:

Er zijn bestanden `spel0.in`, `spel1.in` tot en met `spel5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test1.bat`, die je kunt gebruiken op de volgende manier:

```
test1 niola spel0.in
```

Met deze opdracht test je het programma `niola` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `spel0.in` (of op deze plaats één van de andere bestanden) naar het bestand `spel.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 1. Taakoverzicht.

Opdracht	Programma	Invoer	Uitvoer	Tijdlimiet	Testen	Punten/test	Totaal
1A	<code>niola</code>	<code>spel.in</code>	<code>1a.uit</code>	1 s	5	3	15
1B	<code>niolb</code>	<code>spel.in</code>	<code>1b.uit</code>	1 s	5	4	20
1C	<code>geen</code>	<code>geen</code>	<code>1c.uit</code>	nvt	1	10	10
1D	<code>niold</code>	<code>spel.in</code>	<code>1d.uit</code>	2 s	5	5	25
1E	<code>niole</code>	<code>spel.in</code>	<code>1e.uit</code>	2 s	5	6 (8)	30

Opdracht 1A: Aantal zetten

Schrijf een programma `niola` dat een bestand `spel.in` inleest. Uitvoer is een bestand `1a.uit` van één regel. Hierop staat het aantal verschillende zetten dat in de gegeven beginpositie kan worden gedaan.

Voorbeeld: 4

Opdracht 1B: Van boven naar beneden leeg maken

Schrijf een programma `niob1b` dat een bestand `spel.in` inleest. Uitvoer is een bestand `1b.uit` met op de eerste regel het aantal zetten b dat je kunt spelen als je het volgende voorschrift volgt. Daarna volgen b regels waarop de zetten staan die je moet doen om dat te realiseren.

Voorschrift: Zoek telkens van boven naar beneden en van links naar rechts naar de eerste kaart die je kunt wegnemen. Als je kunt kiezen uit verschillende kaarten die daar bij horen kies je de opnieuw eerste die je van boven naar beneden en van links naar rechts tegen komt.

Voorbeeld: 13
S7H7
S4R4
S5R5
R2H2
K8R8
K3H3
R6S6
STHT
SVRV
HVKV
H9R9
HHRH
RAKA

Opdracht 1C: Hoe snel kun je zo vast komen te zitten?

Schrijf een bestand `1c.uit` dat bestaat uit twee regels.

Op de eerste regel staat `{TASK: 1c}`

Gebruik van andere commentaarhaken is mogelijk.

Op de tweede regel staat een getal k met $0 \leq k \leq 26$ dat staat voor het aantal zetten dat het spel minimaal zal duren voordat er geen geldige zetten meer mogelijk zijn, wanneer je te werk gaat zoals bij opgave 1B.

Let op, je krijgt hiervoor geen invoerbestand, omdat het gaat om het algemene geval. Je mag je zelf een spel voorstellen dat zo ongunstig mogelijk ligt. Als er nog zetten mogelijk zijn moet je die doen. De vraag is dus na hoeveel zetten het op zijn vroegst mogelijk is dat er geen geldige zetten meer zijn.

Opdracht 1D: De eerste rij leegmaken

Schrijf een programma `nioid` dat een bestand `spel.in` inleest. Uitvoer is een bestand `1d.uit` met op de eerste regel het minimale aantal zetten z dat je nodig hebt om de eerste rij z leeg te maken. Daarna volgen z regels waarop de zetten staan die je achtereenvolgens moet doen om dat te realiseren.

Bij alle testgevallen die je krijgt zal het mogelijk zijn deze eerste rij in zijn geheel leeg te maken.

Voorbeeld: 4
S7H7
S4R4
S5R5
R2H2

Opdracht 1E: Zo ver mogelijk uitspelen

Schrijf een programma `nio1e` dat een bestand `spel.in` inleest. Uitvoer is een bestand `1e.uit` met op de eerste regel het aantal zetten `v` dat je maximaal kunt spelen vanuit de gegeven positie. Daarna volgen `v` regels met daarop de benodigde zetten.

Uiteraard probeer je het spel uit te spelen; als dat niet lukt wil je zo weinig mogelijk kaarten overhouden.

De jury heeft voor de verschillende testgevallen met een eenvoudig programma een oplossing gevonden. Als je die oplossing ook vindt krijg je de 6 punten voor de test. Als jouw oplossing één zet minder telt, krijg je nog 2 punten. Mocht je het beter doen dan de jury, dan krijg je zelfs 8 punten als je oplossing klopt.

Voorbeeld:

```
26
S7H7
R8H8
S6K6
S3H3
H9S9
R4H4
KAHA
R2K2
S5R5
K8S8
K3R3
HTRT
RHKH
KBHB
RASA
S4K4
R7K7
SVRV
RBSB
R9K9
R6H6
STKT
HSHH
H5K5
S2H2
HVKV
```