

Opgave 0. Mahjong Rivers

Mahjong Rivers is een patiencespel dat gespeeld wordt met Mahjongstenen. In de beginopstelling liggen de stenen in een rechthoek. Het doel is eenvoudig: verwijder alle stenen van het bord door paren bij elkaar te zoeken. Er is maar één regel om twee identieke stenen met elkaar te verbinden: Als je probeert om een aaneengesloten lijn te trekken van de één naar de ander, een lijn die uitsluitend bestaat uit horizontale en verticale stukken, door de vrije ruimte om of tussen de stenen, dan mogen er niet meer dan twee hoeken in de lijn zitten, dat wil zeggen dat als je in je route meer dan twee keer van richting moet veranderen voor je op de andere steen staat, dan kun je deze stenen (nog) niet verwijderen.



Schermbbeeld van Kyodai Mahjong Rivers (downloaden van www.kyodai.com).

In deze opgave moet je programma's schrijven om enkele vragen bij een stelling in een versimpelde versie van Mahjong Rivers te beantwoorden. Versimpeld, omdat er maar 48 stenen zijn, namelijk 24 paren. De stenen worden aangeduid met de hoofdletters A tot en met X.

Invoer.

De stelling staat in een bestand `spel.in` dat bestaat uit zes regels. Op elke regel staan acht tekens; een hoofdletter om een steen aan te duiden of een min-teken '-' om een lege plaats aan te geven.

Voorbeelden

spel0.in:	ABC---G-	spel1.in	AABBCCDD
	QV-RTU--		EFGHIJKL
	---KM---		EFGHIJKL
	-A---GX-		-----
	QR-TUVXW		QRSTUVWXYZ
	-KWC-B-M		XWVUTSRQ

Deze bestanden worden bij de voorbeelden bij alle opgaven als invoer gebruikt.

A	B	C				G	
Q	V		R	T	U		
			K	M			
	A				G	X	
Q	R		T	U	V	X	W
	K	W	C		B		M

Het voorbeeld van bestand `spel0.in`

Voorbeeldbestanden en testen:

Er zijn bestanden `spel0.in`, `spel1.in` tot en met `spel15.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test0.bat`, die je kunt gebruiken op de volgende manier:

```
test0 nio0a spel0.in
```

Met deze opdracht test je het programma `nio0a` (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.

Opgave 0 overzicht

Onderdeel	Programma	Uitvoer	Tijdlimiet per test	Aantal testen	Punten per test	Totaal
0A	<code>nio0a</code>	<code>0a.uit</code>	1 sec	6	2	12
0B	<code>nio0b</code>	<code>0b.uit</code>	1 sec	6	3	18
0C	<code>nio0c</code>	<code>0c.uit</code>	1 sec	5	6	30
0D	<code>nio0d</code>	<code>0d.uit</code>	1 sec	5	8	40

Onderdeel 0A: Hoeveel stenen liggen er nog?

Schrijf een programma `nio0a` dat een bestand `spel.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `0a.uit` dat bestaat uit één regel. Daarop staat aangegeven uit hoeveel stenen de ingevoerde stelling bestaat.

Uitvoer bij de gegeven voorbeelden: `spel0.in` 26

`spel1.in` 40

Onderdeel 0B: De verwijderde paren.

Schrijf een programma `ni00b` dat een bestand `spe1.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `0b.uit` dat bestaat uit één regel. Op die regel staan de letters aangegeven die niet meer in de stelling voorkomen, op alfabetische volgorde, niet gescheiden door spaties. Als alle stenen nog aanwezig zijn geeft je programma als uitvoer het woord "geen".

Uitvoer bij de gegeven voorbeelden: `spe10.in` DEFHIJLNOPS

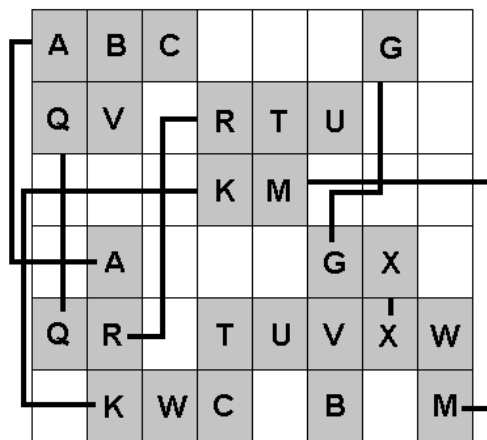
`spe11.in` MNOP

Onderdeel 0C: Te verwijderen paren.

Schrijf een programma `ni00c` dat een bestand `spe1.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `0c.uit` dat bestaat uit één regel. Op die regel staan de letters aangegeven van de paren die in deze stelling verwijderd kunnen worden, op alfabetische volgorde, niet gescheiden door spaties. Als er geen stenen verwijderd kunnen worden geeft je programma als uitvoer het woord "geen".

Uitvoer bij de gegeven voorbeelden: `spe10.in` AGKMQRX

`spe11.in` ABCDEFGHIJKL



Alle te verwijderen paren in de stelling van `spe10.in`

Onderdeel 0D: Telkens de eerst mogelijke zet.

Voor dit onderdeel kiest je programma uit de lijst van paren die kunnen worden verwijderd telkens de eerste in alfabet. Daarna moet het aangeven hoeveel paren zijn overgebleven en welke paren dat zijn.

Schrijf een programma `ni0d` dat een bestand `spel.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `od.out` dat bestaat uit drie regels. Op de eerste regel staan de zetten die in de gegeven stelling worden gedaan; telkens wordt daarbij de eerste zet op alfabet die mogelijk is uitgevoerd. Op de tweede regel staat aangegeven hoeveel paren nu overblijven. Op de derde regel staan deze paren aangegeven, op alfabetische volgorde

Uitvoer bij de gegeven voorbeelden:

<code>spel0.in</code>	AGKMQRTCBUVWX 0
<code>spel1.in</code>	ABCDEFGHIJKL 8 QRSTUVWXYZ