

Lego

Mit Lego-Steinen trainierst du ein Bildverarbeitungssystem. Schreibe ein Programm das für zwei aus verschiedenen Winkeln aufgenommene Bilder einer Lego-Konstruktion berechnet, wie viele auf unterschiedliche Weise erbaute Konstruktionen zu diesen Bildern passen.

In dieser Aufgabe gibt es nur 2x2-Steine in drei verschiedenen Farben: weiß (W), grau (G) oder schwarz (B – für „black“). Es gibt in jeder Farbe beliebig viele Steine. Die Basisplatte ist ein 6x6-Quadrat. In einer Konstruktion müssen die Kanten der Steine natürlich parallel zu den Kanten der Platte liegen und dürfen nicht über die Platte hinausragen. Jeder Stein muss auf mindestens einem Block aufbauen (oder auf der Platte).



Links: ein möglicher Aufbau.

Mitte: unmöglich (der blaue Stein hängt in der Luft).

Rechts: auch unmöglich (der blaue Stein ragt über die Platte hinaus).

Eingabe.

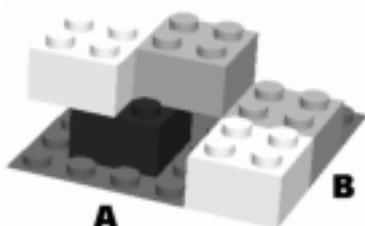
Die erste Zeile der Datei `lego.in` enthält ein Integer H ($1 \leq H \leq 6$), die Höhe der Konstruktion. Dann folgen H Zeilen mit je 6 Zeichen, die das Bild der Konstruktion beschreiben, das sich von einer Seite aus bietet (im Bild unten mit A bezeichnet). Zeichen j in Zeile i gibt also an, was in der j -ten Spalte von links in der i -ten Reihe von oben zu sehen ist. Das Zeichen kann entweder 'W', 'G', 'B' oder '.' sein und gibt eine Farbe an bzw. ein Loch ('.'). Zur Bildtiefe, in der das jeweilige Element sich befindet, gibt es keine Information.

Dem ersten „Bild“ folgt ein zweites (mit ebenfalls H Zeilen); es beschreibt die Ansicht der Konstruktion nach Bewegung des Betrachters um 90 Grad gegen den Uhrzeigersinn (also von der Seite, die in der Abbildung unten mit B bezeichnet ist).

Ausgabe.

Dein Programm soll eine Zeile in die Datei `lego.out` schreiben, mit einem einzigen Integer: Die Anzahl unterschiedlicher Lego-Konstruktionen, die zu den Eingabe-„Bildern“ passen. Du kannst davon ausgehen, dass diese Zahl immer in ein signed 64-bit Integer passt. Achtung: Zwei unterschiedliche Konstruktionen zählen auch dann beide, wenn sie durch Rotation oder Spiegelung ineinander überführt werden können.

Beispiel.	<code>lego.in</code>	<code>lego.out</code>
	2	6
	WWGG..	
	.BB.WW	
	.WGG..	
	WWGG..	



Eine der sechs Konstruktionen, die zur Beispieleingabe passen.