

## Lego

Du använder legoklossar för att träna upp ett AI-bildsystem. Skriv ett program som givet två bilder av en legokonstruktion från två olika vinklar, beräknar hur många olika sätt konstruktionen kan byggas på.

I den här uppgiften, finns bara en sorts legokloss (med 2x2 pluppar, se bild nedan), men den kan ha 3 olika färger, vit (W), grå (G), eller svart (B). Alla klossar finns i obegränsad mängd. Du utgår ifrån en kvadratisk basplatta med 6x6 pluppar. Varje kloss ska vara parallell mot basplattan och ingen kloss får hänga utanför plattan. Varje kloss måste sitta fast i åtminstone en underliggande kloss.



Vänster: Ett tillåtet sätt att placera en kloss ovanpå en annan kloss.

Mitten: Ett otillåtet sätt (den övre klossen hänger i luften).

Höger: Ett otillåtet sätt (den övre klossen sticker utanför basplattan).

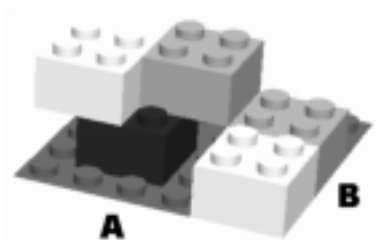
**Indata.** Den första raden i `lego.in` innehåller ett heltal  $H$  ( $1 \leq H \leq 6$ ), som är höjden i konstruktionen. Därefter följer  $H$  rader med 6 tecken på varje rad som ger en bild av konstruktionen sett från en sida (markerad A i figuren under). Tecken nr  $j$  på rad nr  $i$  anger vad du ser på kolumn nr  $j$  (räknat från vänster) på rad nr  $i$  (räknat från ovan). Varje tecken kommer vara antingen 'W', 'G', 'B' (vilket anger färgen) eller '.' vilket anger ett hål. Observera att du inte får reda på djupet, så varje kloss sedd kan antingen vara nära eller långt borta.

Den första bilden följs av ytterligare  $H$  rader som visar konstruktionen sedd från en vinkel där observatören har roterat 90 grader moturs kring konstruktionen (markerad B i figuren nedan).

**Utdata.** Programmet skall skriva en rad till `lego.out` innehållande endast ett heltal: antalet unika legokonstruktioner som skulle ge upphov till bilderna i indatan. Observera att två konstruktioner som inte är helt identiska, men ändå är rotationer eller speglingar av varandra ska räknas som två olika konstruktioner. För den givna indatan kommer svaret alltid få plats i ett 64-bitars signed integer.

**Exempel.**

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



En av de möjliga konstruktionerna från exemplet.