






# HAVEN



## Opgave

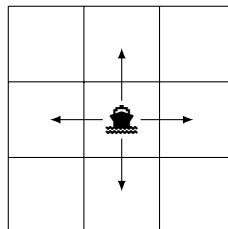
De rastervormige haven bestaat uit zeevakjes en eilandvakjes.

- Een zeevakje stellen we hier voor met .
- Elk eilandvakje bevat een container met goederen van een bepaald type, voorgesteld door een letter. Eenzelfde type goederen kan meermaals voorkomen.

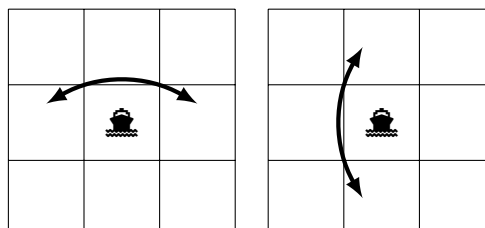
A	C	D
		B
		A

Voorbeeld: een  $3 \times 3$  haven


Wij besturen een bootje en kunnen doorheen de zeevakjes varen. We kunnen enkel in de vier windrichtingen (noord/oost/zuid/west) bewegen, niet diagonaal.



De containers staan echter niet op hun juiste plaats. Gelukkig kunnen we, indien we met ons bootje tussen 2 containers staan, deze van plaats verwisselen. Met behulp van ons bootje wensen we zoveel mogelijk containers op hun juiste plaats te zetten. Containers verwisselen kan op twee manieren:



Het bootje kan een willekeurig aantal containers verwisselen. We vragen ons nu af hoeveel containers er *niet* op hun juiste plek kunnen eindigen.

A	≡	B	X
	H	≡	I
≡	≡	≡	A
Y	D	≡	E

→

B	≡	A	X
≡	I	≡	H
≡	≡	≡	Y
A	E	≡	D

- De containers H en I kunnen van plaats verwisseld worden. Zo ook de containers D en E.
- A en B in de bovenste rij kunnen *niet* verwisseld worden: het bootje geraakt niet op de tussenliggende positie.
- A en Y in de linkerkolom kunnen tevens niet verwisseld worden: hiervoor moet exact één zeevakje tussen beide containers liggen (en het bootje moet daar ook geraken).

Het beste dat we kunnen verkrijgen is

A	≡	B	X
≡	I	≡	H
≡	≡	≡	A
Y	E	≡	D

Met andere woorden, er zijn vier containers waarvoor we er niet voor kunnen zorgen dat deze op hun doelpositie belanden.

Enkele extra verduidelijkingen:

- De begin- en doeltoestand vertonen gegarandeerd dezelfde structuur: zeevakjes blijven zeevakjes en eilandvakjes blijven eilandvakjes. Enkel het goederentype kan verschillen.
- Begin- en doeltoestand bevatten elk hetzelfde aantal van elk goederentype. Bv. indien we beginnen in een haven met 5 containers van type A, zal de doeltoestand eveneens 5 containers van type A bevatten.
- Een container die initieel reeds op zijn doelpositie staat moet uiteraard niet meegeteld worden in het eindresultaat.

## Invoer

De eerste regel bevat een positief geheel getal dat het aantal testgevallen voorstelt. Per testgeval volgt:

- Een regel met twee positieve gehele getallen  $B$  en  $H$  die de breedte en hoogte voorstellen van de haven.
- $H$  regels met telkens  $B$  tekens die de begintoestand van de haven voorstellen.
- $H$  regels met telkens  $B$  tekens die de doeltoestand voorstellen.

Hierbij geldt

- `.` stelt een zeevakje voor.
- hoofdletters van A tot Z stellen een type container voor
- `*` geeft de startpositie voor van het bootje.

---

### VOORBEELDINVOER

---

```
3
3 3
*.A
...
C.B
..A
...
C.B
4 4
A.D.
B.C.
....
*.E.
B.D.
A.E.
....
..C.
5 3
A.X.B
..Y..
*.Z..
B.Z.A
..Y..
..X..
```

---

## Uitvoer

Per testgeval voer je een regel uit met twee gehele getallen  $I$  en  $N$ :

- $I$  is de index van het testgeval. We beginnen te tellen bij 1.
- $N$  geeft aan hoeveel containers er *niet* op hun doelpositie kunnen eindigen.

---

### VOORBEELDUITVOER

---

1	0
2	2
3	4

---