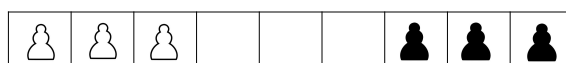


# Spring en schuif

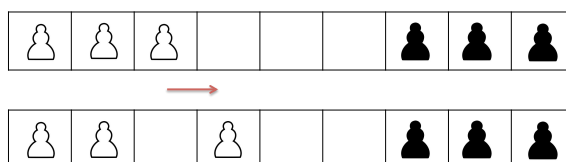


Het spel ‘Spring en schuif’ wordt gespeeld op een ééndimensionaal spelbord. Er zijn twee spelers. De ene speler speelt met witte pionnen, de andere met zwarte. Het spelbord bestaat uit een rij vakjes. Elk vakje kan hoogstens 1 pion bevatten. Het spel start met alle witte pionnen links, en alle zwarte pionnen rechts, zoals in Figuur 1.

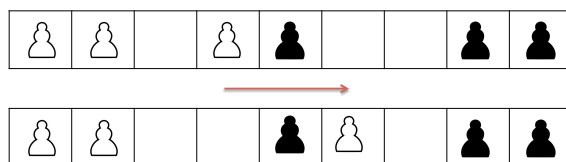


Figuur 1: beginsituatie

Speler ‘wit’ kan slechts 2 soorten zetten doen. Hij kan een witte pion één vakje naar rechts verplaatsen (**schuif**), dit kan echter alleen als het vakje rechts van deze pion leeg is. De speler kan ook een witte pion over een zwarte naar een leeg vakje verplaatsen (**spring**), dit kan enkel als de witte pion zich vlak naast een zwarte pion bevindt, en er een leeg vakje is aan de andere kant van de zwarte pion. Deze beweging kan ook enkel naar rechts uitgevoerd worden. De zet **schuif** wordt voorgesteld in Figuur 2, de zet **spring** in Figuur 3. Speler ‘zwart’ kan analoge zetten uitvoeren, maar dan in de tegenovergestelde richting.



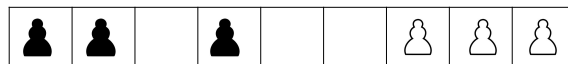
Figuur 2: de zet **schuif**



Figuur 3: de zet **spring**

De speler met de witte pionnen mag altijd beginnen, daarna wordt steeds afwisselend een zet gedaan. Men mag niet passen. Indien een speler geen zet kan uitvoeren, eindigt het spel. Speler ‘wit’ wint indien hij bij het einde van het spel al zijn witte pionnen uiterst rechts heeft staan, zonder dat daar

nog zwarte pionnen tussen staan, zoals bijvoorbeeld in Figuur 4. Als dat niet het geval is, dan wint speler ‘zwart’, zoals bijvoorbeeld in Figuur 5.



Figuur 4: eindsituatie waarbij speler ‘wit’ wint



Figuur 5: eindsituatie waarbij speler ‘zwart’ wint

## Opgave

Voor deze opgave gaan we slechts een deeltje van dit spel implementeren, namelijk de module die nakijkt welke zetten mogelijk zijn voor een bepaalde speler vanuit een gegeven toestand van het spelbord.

Een spelbord wordt voorgesteld door een rij letters: WWLLLZZZ. W stelt een vakje met een witte pion voor, Z een vakje met een zwarte pion en L een leeg vakje.

Bijvoorbeeld:

- Gegeven WWLLLZZZ en speler ‘wit’ is aan de beurt, dan zijn er geen **spring** zetten mogelijk, er is wel 1 **schuif** zet mogelijk.
- Gegeven WLWZLZWZ en speler ‘zwart’ is aan de beurt, dan zijn er 2 **spring** zetten mogelijk en 1 **schuif** zet.
- Gegeven WZZLLLWWZ en speler ‘zwart’ is aan de beurt, dan zijn er 0 **spring** zetten mogelijk en 0 **schuif** zetten<sup>1</sup>.

## Invoer

Aan de invoer komt eerst een regel met een getal  $n$  dat aangeeft hoeveel speltoestanden er volgen ( $1 \leq n \leq 1000$ ). Daarna volgen  $n$  regels met speltoestanden. Een speltoestand wordt voorgesteld door een letter die aangeeft welke speler aan zet is (W of Z), daarna komt een spatie, gevolgd door de toestand van het spelbord.

Merk op dat de grootte van het bord niet noodzakelijk 9 is zoals in de voorbeeldjes. Je mag er wel van uit gaan dat het bord niet groter is dan 1000 vakjes. Er kunnen ook meer dan 3 witte en zwarte pionnen zijn, er zijn niet noodzakelijk evenveel witte als zwarte pionnen.

<sup>1</sup>Het spel is hierbij ook ten einde en speler ‘zwart’ is gewonnen.

## Uitvoer

De uitvoer van het programma bevat  $n$  regels. Elke regel bevat twee getallen, gescheiden door een spatie. Het eerste getal komt overeen met het aantal mogelijke **spring** zetten, het tweede getal geeft het aantal mogelijke **schuif** zetten weer. De volgorde van de regels komt overeen met de volgorde waarin de betreffende speltoestanden aan de invoer verschijnen.

Let op! Zorg ervoor dat je uitvoer geen overbodige tekens bevat, bijvoorbeeld een spatie op het eind van een regel of een lege regel op het eind van de uitvoer. Dat zorgt er immers voor dat je uitvoer als foutief wordt beschouwd.

## Voorbeeld

### Invoer

```
3
W WWWLLLZZZ
Z WLWZLZLWZ
Z WZZLLLWWZ
```

### Uitvoer

```
0 1
2 1
0 0
```