

# Sokoban



## Opgave

Sokoban is een spel waarbij een *mijnwerker* blokjes moet verplaatsen in een tweedimensionaal rooster teneinde deze blokjes in vooraf vastgelegde doelposities te plaatsen. In een rooster zijn er, behalve *lege posities*, ook *blokjes*, *muren* en *doelposities*.

De mijnwerker kan slechts één blokje tegelijkertijd verplaatsen en dit door te duwen tegen het blokje. (De mijnwerker kan dus niet trekken aan blokjes.) De mijnwerker kan niet op of door muren lopen. De mijnwerker kan niet op een blokje staan. De mijnwerker kan wel over doelposities lopen. Je mag ervan uitgaan dat de mijnwerker in een gesloten rooster werkt, d.w.z. dat aan de rand van het rooster (dat niet noodzakelijk rechthoekig is) muren staan.

De mijnwerker kan in principe horizontaal en verticaal bewegen. Dit lukt als het vakje waarnaar de mijnwerker wil bewegen geen muur is of als het een blokje is en het vakje naast het blokje in dezelfde richting een leeg vakje of een lege doelpositie is. De mijnwerker kan met andere woorden één blokje tegelijkertijd verplaatsen, op voorwaarde dat het vakje waarnaar het blokje wordt verplaatst geen muur of blokje is.

In Figuur 1 zie je een screenshot uit een DOS-versie (uit 1984) van dit spel. In dit voorbeeld kan de mijnwerker naar boven bewegen, waarbij het blokje vooruit wordt geduwd. De mijnwerker kan zich ook verplaatsen naar de lege plaats aan zijn rechterkant. De mijnwerker kan niet naar onder bewegen, want daar staat een muur. De mijnwerker kan ook niet naar links bewegen, want dan zou hij twee blokjes tegelijkertijd moeten verplaatsen. Wanneer één van deze twee laatste bewegingen (beneden/links) wordt uitgevoerd, dan gebeurt er eenvoudigweg niets op het spelbord.

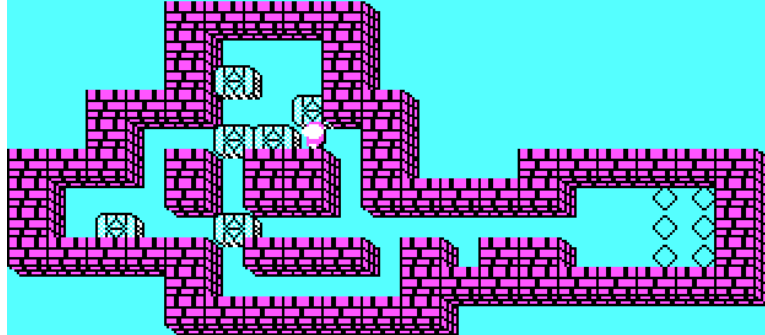
Het doel van het programma is om de bewegingen van de mijnwerker te volgen en de finale configuratie uit te schrijven.

## Invoer

De eerste regel bevat het aantal opgaves  $n$  ( $1 \leq n \leq 100$ ).

Elke opgave bestaat uit  $1 + r + 1 + 1$  regels. De eerste regel bevat de getallen  $r$  en  $c$  ( $1 \leq r, c \leq 40$ ) gescheiden door een spatie. Deze getallen geven respectievelijk het aantal rijen en kolommen waaruit een spelbord bestaat. De volgende  $r$  regels bevatten elk exact  $c$  karakters en beschrijven het spelbord. Daarna volgt een regel met het aantal commando's  $a$  ( $1 \leq$

Figuur 1: Een voorbeeld van een Sokoban spel



$a \leq 1000$ ). Daarna volgt een regel met lengte  $a$  die de commando's bevat. Deze commando's worden van links naar rechts uitgevoerd.

We gebruiken de volgende symbolen om een spelbord te beschrijven:

- ' ' (spatie) is een leeg vakje
- '#' is een muur
- '.' is een doelpositie
- '\$' is een blokje (op een leeg vakje)
- '@' is de mijnwerker (op een leeg vakje)
- '+' is de mijnwerker op een doelpositie
- '\*' is een blokje op een doelpositie

Het spelbord uit Figuur 1 wordt dus als volgt voorgesteld:

```
#####
#  #
#$  #
### $$$
# $$$@ #
### # ## # #####
# # ## ##### ..#
# $ $ ..#
##### ### # ## ..#
# #####
#####
```

De commando's bestaan uit de volgende letters:

- 'L' beweegt de mijnwerker naar links (indien mogelijk)
- 'R' beweegt de mijnwerker naar rechts (indien mogelijk)
- 'B' beweegt de mijnwerker naar boven (indien mogelijk)

- 'O' beweegt de mijnwerker naar onder (indien mogelijk)

Wanneer een commando niet kan uitgevoerd worden (bv. omdat de mijnwerker anders op een muur zou lopen) dan heeft dit geen effect en blijft de mijnwerker gewoon staan. Het spelbord verandert dan niet.

## Uitvoer

Voor elke opgave bestaat de uitvoer uit het resulterende spelbord. Tussen elke twee spelborden staat een lege lijn.

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

## Voorbeeld

### Invoer

```
2
3 5
#####
#O$.#
#####
2
RL
8 6
  ####
  #   #
#   #
# #O #
# $* #
# .* #
#   #
#####
7
OLROLOL
```

## Uitvoer

```
#####  
#@ *#  
#####
```

```
#####  
# #  
# #  
# # #  
# $* #  
# *+ #  
# #  
#####
```