

## Z – Sokoban

If you are familiar with the Sokoban video game, the description of the task is quite short. You have to find the minimum number of box pushes to solve a given puzzle. The number of moves the player makes is not important.

The game Sokoban takes place on a grid, where in every step the player can move to one of the four adjacent squares in either the horizontal or vertical direction. Some squares are blocked by walls and some other squares are occupied by boxes. Certain squares are designated as storage locations. The player has to push boxes onto those storage locations. The player pushes a box by moving into its square, which pushes it into the square beyond. There must be a free square beyond the box, otherwise it cannot be moved. The player also cannot push boxes into other boxes or walls, and boxes cannot be pulled or transferred in any other way. The only allowed method of moving boxes is by pushing them. Even if the box is already on the storage location, it can still be moved. The only important thing is that all boxes end up on the storage locations in the end.

Even very small configurations can be extremely complex, so we need to use an efficient algorithm for solving Sokoban puzzles.

### Input data

A grid of dimensions  $N \times M$  is described in  $N$  lines with at most  $M$  characters. Some lines may contain less than  $M$  characters as they do not contain trailing spaces. It is guaranteed that the player is in a region that is enclosed with the walls. The number of boxes  $B$  equals the number of storage locations. The following list describes the meaning of each character that can be used to describe a puzzle:

```
'#' - a wall
' ' - free square
'.' - free storage location
'@' - the player standing on a square that is not a storage location
'+' - the player standing on a storage location
'$' - a box placed on a square that is not a storage location
'*' - a box placed on a storage location
```

### Input limits

- $3 \leq N, M \leq 8$
- $1 \leq B \leq 4$

### Output data

Output a single line that contains the minimum number of box pushes that are needed to solve the given puzzle. It is guaranteed that a puzzle is solvable.

### Examples

#### Input

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

#### Output

9

#### Input

#### Output

13

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

Input

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

Output

```
6
```