6. Frogger

Program Name: Frogger.java Input File: frogger.dat

Wes is writing a computer application that involves a tadpole getting out of a maze of walls. The tadpole is given an initial starting position in a rectangular maze, which for the purpose of this program is depicted as an `r` x `c` matrix of cells. The tadpole can swim from one cell to a horizontally or vertically adjacent cell, avoiding the walls, until he gets out of the maze by landing on the exit cell.

However, if the tadpole moves into a "magic" cell (a cell denoted by a single digit *d*), he temporarily turns into a frog. A frog has special frog powers that enable him to jump either horizontally or vertically over *d* number of cells and/or walls. If, after he jumps, the frog

- lands on the exit cell, he gets out of the maze.
- lands on an empty cell, he turns back into a tadpole and continues.
- lands on another magic cell, he remains a frog and can jump to another cell if possible.

If the frog cannot jump to one of the above three kinds of cells, he returns to his tadpole state with no magic powers and can swim to another cell if possible.

The starting cell is considered to be the first cell visited by the tadpole. A cell is considered visited when either the tadpole or frog enters it. Neither the tadpole nor the frog

- can visit or jump to a previously visited cell.
- can visit or jump to a wall.
- can go outside the maze.

Input

The first line of input will contain a single integer `n` that indicates the number of mazes to follow. For each maze, the first line will contain the size of the maze in the form `r c` where $3 \le r \le 8$ is the number of rows and $3 \le c \le 8$ is the number of columns in the maze. The next `r` lines will contain `c` characters as follow:

- '**`*`**' is a wall
- '**`.`**' is an empty cell to which the tadpole or frog can move.
- `d` is a digit in the range 1 through 4 which indicates that the tadpole has turned into a frog. `d` is the number of cells (including wall cells) that the frog can jump over, if possible, before turning back into a tadpole. For example, if `d` = 3, the frog would land on the 4th cell from where he started.
- '**`$`**' is the tadpole's starting position
- '**`@`**' is the cell that the tadpole will use to exit the maze

Output

For each maze, you will print the minimum number of cells in the maze that the tadpole will visit before exiting the maze.

Note: Some mazes could take an extraordinarily long time to solve. The data sets for this problem have been designed to execute in less than 5 seconds. Solutions taking longer than 30 seconds to execute will be rejected.

6. Frogger (cont.)

Example Input File

```
2
8 8
******..
....$...
..***.2.
.3..***.
**.2.***
***.....
....@...
****..**
5 7
****$**
**....2
.*..3..
.*....*
@..****
```

Example Output to Screen

```
8
6
```