

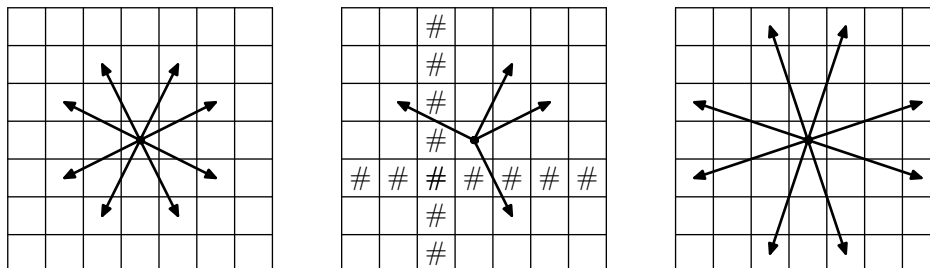
3. Superknight (ratsu)

1 sec

40 points

A superknight is located on some square of a chessboard and wants to move to another square, and do so in the least possible number of moves. It is hindered by the fact that some squares are blocked and the knight is not allowed to visit them on its path.

A regular knight can move either by two rows and one column or by two columns and one row in each move (the leftmost figure). A knight can jump over blocked squares, only landing on them is forbidden (the middle figure).



A superknight differs from a regular one by its ability to perform a *supermove* on its path (but *at most once*), in which it moves either by three rows and one column or by three columns and one row (the rightmost figure).

Write a program to find the path of a superknight from a given starting square to a given destination square with the least possible number of moves.

Input. The first line of the file `ratsusis.txt` contains R , the number of rows, and V , the number of columns of the board ($3 \leq R \leq 100$, $3 \leq V \leq 100$). Each of the following R lines contains exactly V characters, where '@' denotes the starting square of the knight, '*' the destination square, '.' a free square, and '#' a blocked square.

Output. The first line of the file `ratsuval.txt` should contain K , the minimal number of moves needed to reach the destination square. Each of the following $K + 1$ lines should contain two integers r_i and v_i : the row and column numbers of the squares that the knight visits on its path, in the order they are visited. The rows of the board are numbered $1 \dots R$ from top to bottom and the columns $1 \dots V$ from left to right. You may assume that the knight can reach the destination square in all test cases. If there are several paths with the minimal number of moves, output any one of them.

Example.	<code>ratsusis.txt</code>	<code>ratsuval.txt</code>
	4 5	4
	*....	4 1
	2 2
	..#..	4 3
	@....	2 4
		1 1

In this example a supermove is needed to achieve the minimal number of moves. Using only regular moves, at least 5 moves would be needed. If the knight could use the supermove more than once, it could reach the destination in just 3 moves. The path $(4, 1) \rightarrow (3, 3) \rightarrow (1, 2) \rightarrow (2, 4) \rightarrow (1, 1)$ would not be valid, because the knight would visit the blocked square $(3, 3)$.

Grading. In each test, finding the minimal number of moves gives 50% of points and finding the corresponding path gives the remaining 50%. A program that outputs a non-minimal valid

path with its correct length will get 50% of points of the test. In tests worth 20 points in total, supermoves are not needed to reach the destination in the minimal number of moves.