

DFS, BFS On Grid

Bài 1. Tìm đường 1.

Cho một bảng $S[][]$ kích thước $N \times M$, bao gồm các ô trống, các vật cản. Mỗi lần di chuyển bạn có thể di chuyển tới ông cùng cạnh và không có vật cản. Ban đầu bạn ở vị trí điểm S, hãy xác định xem bạn có thể di chuyển tới điểm T được hay ko?

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test bắt đầu bởi hai số nguyên N và M ($1 \leq N, M \leq 500$).

N dòng tiếp theo, mỗi dòng gồm M kí tự mô tả bảng S . Trong đó: ‘.’ là một ô trống, ‘*’ là vật cản, ‘S’ là vị trí xuất phát và ‘T’ là vị trí đích.

Output:

Với mỗi test, in ra “YES” nếu tìm được đường đi, ra in “NO” trong trường hợp ngược lại.

Ví dụ:

Input:	Output
1 5 5 ..S.. ****. T.... ****. 	YES

Bài 2. Labyrinth.

Link Submit : <https://cses.fi/problemset/task/1193>

You are given a map of a labyrinth, and your task is to find a path from start to end. You can walk left, right, up and down.

Input

The first input line has two integers n and m : the height and width of the map.

Then there are n lines of m characters describing the labyrinth. Each character is . (floor), # (wall), A (start), or B (end). There is exactly one A and one B in the input.

Output

First print "YES", if there is a path, and "NO" otherwise.

If there is a path, print the length of the shortest such path and its description as a string consisting of characters L (left), R (right), U (up), and D (down). You can print any valid solution.

Constraints

$$1 \leq n, m \leq 1000$$

Example

Input:

```
5 8
#####
#.A#...#
#.#.#B#
#.....#
#####
```

Output:

```
YES
9
LDDRRRRRU
```

Source tham khảo : <https://ideone.com/LJLQ9A>

Bài 3. Count Island Problem.

Cho một bản đồ kích thước $N \times M$ được mô tả bằng ma trận $A[i][j]$. $A[i][j] = 1$ có nghĩa vị trí (i, j) là nổi trên biển. 2 vị trí (i, j) và (x, y) được coi là liên nhau nếu như nó có chung đỉnh hoặc chung cạnh. Một hòn đảo là một tập hợp các điểm (i, j) mà $A[i][j] = 1$ và có thể di chuyển giữa hai điểm bất kì trong đó.

Nhiệm vụ của bạn là hãy đếm số lượng đảo xuất hiện trên bản đồ.

Input: Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test bắt đầu bởi 2 số nguyên N và M ($1 \leq N, M \leq 500$).

N dòng tiếp theo, mỗi dòng gồm M số nguyên $A[i][j]$.

Output: Với mỗi test, in ra số lượng hòn đảo tìm được.

Ví dụ:

Input:	Output
1 5 5 1 1 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 1	5

Bài 4. Counting Room.

Submit Link : <https://cses.fi/problemset/task/1192>

You are given a map of a building, and your task is to count the number of its rooms. The size of the map is $n \times m$ squares, and each square is either floor or wall. You can walk left, right, up, and down through the floor squares.

Input

The first input line has two integers n and m : the height and width of the map.

Then there are n lines of m characters describing the map. Each character is either `.` (floor) or `#` (wall).

Output

Print one integer: the number of rooms.

Constraints

$$1 \leq n, m \leq 1000$$

Example

Input:

5 8

#####

#.#...#

####.#.#

#.#...#

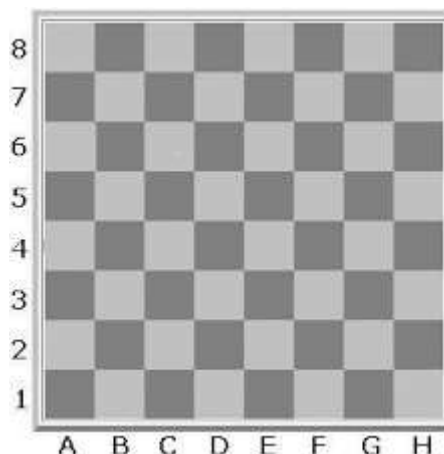
#####

Output:

3

Bài 5. Quân mã.

Cho một quân mã trên bàn cờ vua tại vị trí ST. Nhiệm vụ của bạn là hãy tìm số bước di chuyển ít nhất để đưa quân mã tới vị trí EN.



Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm 2 xâu dạng “xy” và “uv”, trong đó x, y là kí tự trong “abcdefgh” còn u, v là số thuộc 1, 2, 3, 4, 5, 6, 7, 8.

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

Ví dụ:

Input:	Output
8	2
e2 e4	4
a1 b2	2
b2 c3	6
a1 h8	5
a1 h7	6
h8 a1	1
b1 c3	0
f6 f6	

Bài 6. Tìm đường 2.

Cho một bảng $S[][]$ kích thước $N \times M$, bao gồm các ô trống, các vật cản. Ban đầu bạn ở vị trí S. Nhiệm vụ của bạn là hãy di chuyển tới vị trí T, sao cho số lần đổi hướng không quá hai lần.

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test bắt đầu bởi hai số nguyên N và M ($1 \leq N, M \leq 500$).

N dòng tiếp theo, mỗi dòng gồm M kí tự mô tả bảng S. Trong đó: ‘.’ là một ô trống, ‘*’ là vật cản, ‘S’ là vị trí xuất phát và ‘T’ là vị trí đích. (Chỉ có một vị trí S và T duy nhất).

Output:

Với mỗi test, in ra “YES” nếu tìm được đường đi, ra in “NO” trong trường hợp ngược lại.

Ví dụ:

Input:	Output
2	
5 5	
..S..	
****.	
T....	
****.	YES
.....	NO
5 5	
S....	
****.	
.....	
.****	
..T..	

Bài 7. Di chuyển tránh vật cản.

Cho một bảng kích thước $N \times N$, trong đó có các ô trống '.' và vật cản 'X'. Các hàng và các cột được đánh số từ 0.

Mỗi bước di chuyển, bạn có thể đi từ ô (x, y) tới ô (u, v) nếu như 2 ô này nằm trên cùng một hàng hoặc một cột, và không có vật cản nào ở giữa.

Cho điểm xuất phát và điểm đích. Bạn hãy tính số bước di chuyển ít nhất?

Input:

Dòng đầu ghi số bộ test (không quá 10). Mỗi test gồm:

Dòng đầu tiên là số nguyên dương N ($1 \leq N \leq 100$).

N dòng tiếp theo, mỗi dòng gồm N kí tự mô tả bảng.

Cuối cùng là 4 số nguyên a, b, c, d với (a, b) là tọa độ điểm xuất phát, (c, d) là tọa độ đích. Dữ liệu đảm bảo hai vị trí này không phải là ô có vật cản.

Output:

Với mỗi test, in ra một số nguyên là đáp số của bài toán.

Ví dụ:

Input	Output
1 3 .X. .X. ...	3

0 0 0 2	
---------	--

Bài 8. Di chuyển trong không gian.

Cho một hình hộp chữ nhật có kích thước $A \times B \times C$, trong đó A là chiều cao, B là chiều rộng và C là chiều dài. Mỗi ô có thể là một ô trống '.' hoặc vật cản '#'.

Mỗi bước, bạn được phép di chuyển sang một ô kề bên cạnh (không được đi chéo). Nhiệm vụ của bạn là tìm đường đi ngắn nhất bắt đầu 'S' tới vị trí kết thúc 'E'.

Input:

Dòng đầu tiên là số lượng bộ test T ($1 \leq T \leq 50$).

Mỗi test bắt đầu bởi 3 số nguyên A, B, C ($A, B, C \leq 30$).

Tiếp theo là A khối, mỗi khối gồm $B \times C$ kí tự mô tả một lát cắt của hình hộp chữ nhật. Giữa 2 khối có một dấu xuống dòng.

Output:

In ra một số nguyên là đường đi ngắn nhất từ S tới E. Nếu không di chuyển được, in ra -1.

Ví dụ:

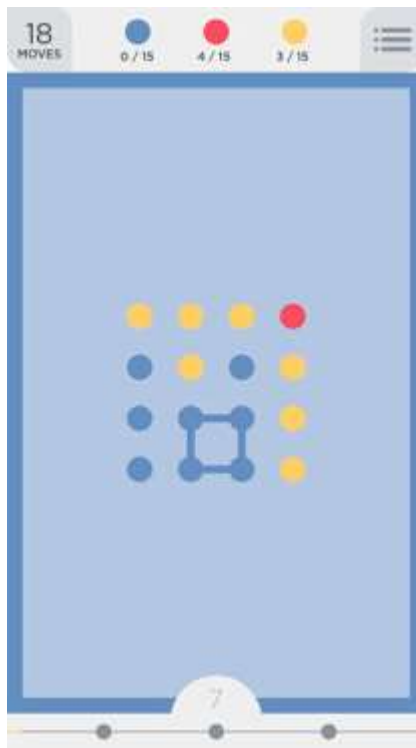
Input	Output
2	
3 4 5	
S....	
.###.	11
.##..	-1
###.#	
#####	

#####	
##.##	
##...	
#####	
#####	
#.###	
####E	
1 3 3	
S##	
#E#	
###	

Bài 9. Fox and two dot

Link submit : <https://codeforces.com/problemset/problem/510/B>

Fox Ciel is playing a mobile puzzle game called "Two Dots". The basic levels are played on a board of size $n \times m$ cells, like this:



Each cell contains a dot that has some color. We will use different uppercase Latin characters to express different colors.

The key of this game is to find a cycle that contain dots of same color. Consider 4 blue dots on the picture forming a circle as an example. Formally, we call a sequence of dots d_1, d_2, \dots, d_k a cycle if and only if it meets the following condition:

1. These k dots are different: if $i \neq j$ then d_i is different from d_j .
2. k is at least 4.
3. All dots belong to the same color.
4. For all $1 \leq i \leq k - 1$: d_i and d_{i+1} are adjacent. Also, d_k and d_1 should also be adjacent.

Cells x and y are called adjacent if they share an edge.

Determine if there exists a cycle on the field.

Input

The first line contains two integers n and m ($2 \leq n, m \leq 50$): the number of rows and columns of the board.

Then n lines follow, each line contains a string consisting of m characters, expressing colors of dots in each line. Each character is an uppercase Latin letter.

Output

Output "Yes" if there exists a cycle, and "No" otherwise.

Examples

Input	Output
3 4 AAAA ABCA AAAA	Yes
3 4 AAAA ABCA AADA	No
4 4 YYR BYBY BBBY BBBY	Yes
7 6 AAAAAB ABBBAB ABAAAB ABABBB ABAAAB ABBBAB AAAAAB	Yes
2 13 ABCDEFGHIJKLM NOPQRSTUVWXYZ	No

Bài 10. Mazes

Link submit : <https://codeforces.com/problemset/problem/377/A>

Pavel loves grid mazes. A grid maze is an $n \times m$ rectangle maze where each cell is either empty, or is a wall. You can go from one cell to another only if both cells are empty and have a common side.

Pavel drew a grid maze with all empty cells forming a connected area. That is, you can go from any empty cell to any other one. Pavel doesn't like it when his maze has too little

walls. He wants to turn exactly k empty cells into walls so that all the remaining cells still formed a connected area. Help him.

Input

The first line contains three integers n, m, k ($1 \leq n, m \leq 500, 0 \leq k < s$), where n and m are the maze's height and width, correspondingly, k is the number of walls Pavel wants to add and letter s represents the number of empty cells in the original maze.

Each of the next n lines contains m characters. They describe the original maze. If a character on a line equals ".", then the corresponding cell is empty and if the character equals "#", then the cell is a wall.

Output

Print n lines containing m characters each: the new maze that fits Pavel's requirements. Mark the empty cells that you transformed into walls as "X", the other cells must be left without changes (that is, "." and "#").

It is guaranteed that a solution exists. If there are multiple solutions you can output any of them.

Examples

Input	Output
3 4 2 #..# ..#. #...	#.X# X.#. #...
5 4 5 #... #.#. .#.. ...# .#.#	#XXX #X#. X#.. ...# .#.#