

Đường đi rô-bốt

Bạn vừa tạo ra một bảng để cho rô-bốt có thể tìm đường đi từ ô ở trên cùng – bên trái (ô xuất phát) đến ô ở dưới cùng – bên phải (ô đích). Tuy nhiên, do quên mất một số skill AI mà bạn chỉ lập trình cho rô-bốt có thể đi sang phải 1 ô hoặc xuống dưới 1 ô. Bạn đặt một số chướng ngại vật trên các ô của bảng (dĩ nhiên là rô-bốt ko thể đi vào các ô này), sau đó bạn ngồi quan sát. Tuy nhiên, sau một thời gian, bạn cảm thấy mệt mỏi vì nó bị mắc kẹt và bạn tự hỏi: “Có bao nhiêu đường đi có thể cho rô-bốt từ ô xuất phát tới ô đích” và “Nếu không có, thì liệu rô-bốt có thể đến ô đích nếu nó được lập trình có thể đi lên trên 1 ô và sang trái 1 ô”.

Vì vậy, bạn quyết định viết 1 chương trình, cho kích thước của bảng $n \times n$ với các chướng ngại vật đã được đánh dấu mà rô-bốt không thể đi tới. Đếm số đường đi khác nhau mà rô-bốt có thể đi từ ô xuất phát tới ô đích. Và nếu không có đường đi, bạn phải kiểm tra xem có thể đi từ ô xuất phát tới ô đích nếu có thể sang trái và lên trên. Tuy nhiên, chương trình của bạn không xử lý các số rất lớn, do đó, kết quả phải được lấy dư cho $2^{31}-1$.

Dữ liệu:

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 1000$).

N dòng sau, mỗi dòng chứa n kí tự đại, mỗi kí tự diện cho một ô của bảng. Kí tự có thể là '.' hoặc '#'. Kí tự '.' nếu ô đó có thể đi, hoặc '#' nếu ô đó là chướng ngại vật. Không có trường hợp có chướng ngại vật ở ô xuất phát và ô đích.

Kết quả:

In ra một dòng chứa số nguyên là số đường đi khác nhau từ ô xuất phát tới ô kết thúc (lấy dư cho $2^{31}-1$) hoặc “THE GAME IS A LIE” nếu không thể đi từ ô xuất phát tới ô kết thúc bằng cách chỉ sang phải và xuống dưới nhưng có thể đi nếu chấp nhận thêm cách đi lên trên và sang trái, hoặc INCONCEIVABLE nếu đơn giản là không có đường đi từ ô xuất phát tới ô đích.

Ví dụ:

INPUT	OUTPUT
5	6
.....	
#..#.	
#..#.	
...#.	
.....	

INPUT	OUTPUT
-------	--------

7	THE GAME IS A LIE
.....#	
####...	
#.....	
#...#.	
#.....	
#..###	
#.....	