

Tấn công đôi

Cờ vua là một trong những trò chơi phổ biến nhất thế giới với hàng triệu người trên toàn thế giới.

Trò chơi sử dụng một bàn cờ hình vuông chia thành 64 ô vuông nhỏ hơn với 8 hàng ngang và 8 hàng dọc.

Thông thường sẽ dùng các số từ 1 → 8 và các chữ cái a → h để biểu diễn hàng cột của bàn cờ, tuy nhiên trong bài toán này, để đơn giản chúng ta chỉ dùng các số từ 1 → 8 để biểu diễn hàng và cột của bàn cờ.

Trong cờ vua, đòn phổ biến nhất là **đòn tấn công đôi**. Đòn tấn công đôi là đòn chiến thuật dùng một quân của bạn để tấn công vào nhiều mục tiêu cùng lúc (thường là 2 mục tiêu).

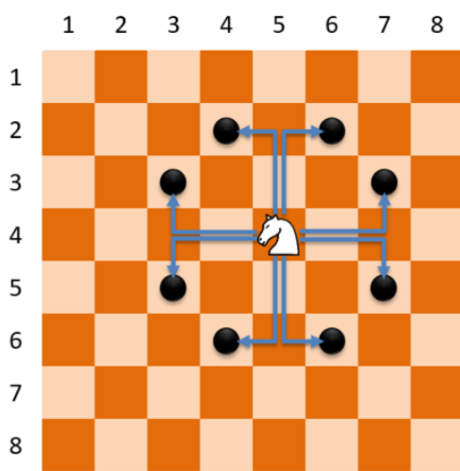
Do việc phòng thủ nhiều mục tiêu cùng lúc rất khó khăn, nên đòn tấn công đôi thường sẽ mang lại lợi thế hơn quân cho kỳ thủ.

Hai quân cờ phổ biến dùng để tấn công đôi là Hậu và Mã. Trong bài toán này chúng ta sẽ chỉ đề cập đến **tấn công đôi bằng Mã**.

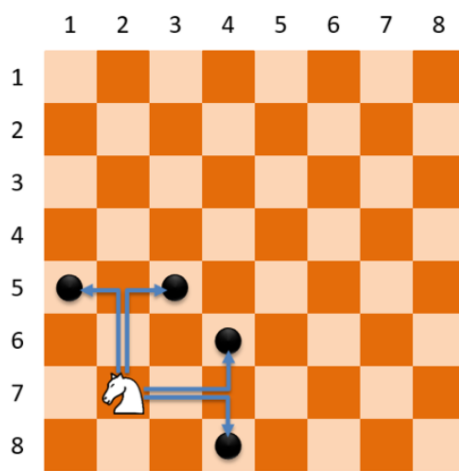
Quy tắc di chuyển (tấn công) của quân Mã như **Hình 1** dưới đây, từ một vị trí quân Mã có thể di chuyển theo hình chữ L kích thước 2 x 3 tới tối đa 8 ô khác.

Các quân không được di chuyển ra khỏi bàn cờ, do đó ở một số vị trí khả năng di chuyển của quân Mã sẽ bị hạn chế, nhất là ở các vị trí góc bàn cờ, ví dụ trong **Hình 2**, tại vị trí hàng 7, cột 2 (7, 2), quân Mã chỉ có khả năng di chuyển tới 4 ô sau lần đi.

Khả năng tấn công của Mã



Hình 1

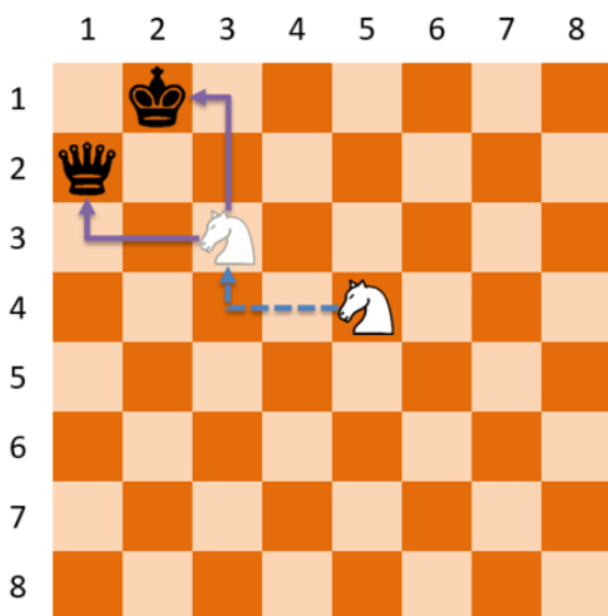


Hình 2

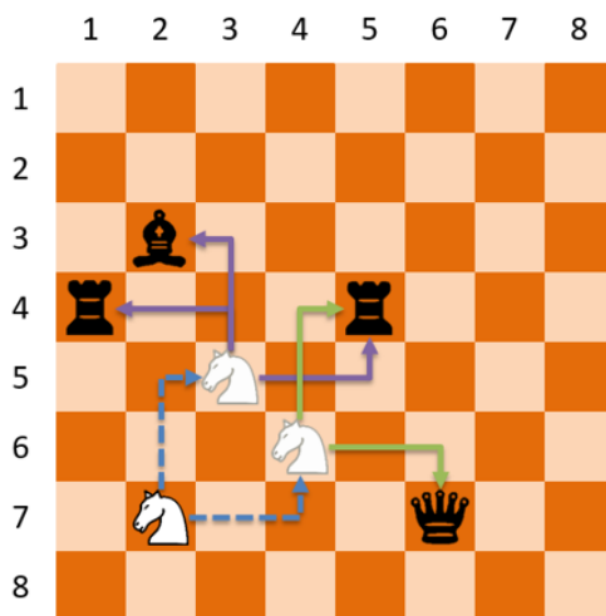
Để thực hiện đòn tấn công đôi, quân Mã cần di chuyển đến một vị trí mà ở vị trí đó nó có thể tấn công vào ít nhất 2 quân của đối thủ. Một quân của đối thủ bị tấn công nếu như nó nằm tại vị trí mà quân Mã có thể di chuyển tới ở những lượt đi tiếp theo.

Trong **Hình 3** dưới đây (test case #1), từ vị trí (4, 5) quân Mã cần di chuyển tới vị trí (3, 3) để thực hiện đòn tấn công đôi vào Vua (1, 2) và Hậu (2, 1).

Tấn công đôi bằng Mã



Hình 3



Hình 4

Trong Hình 4 (test case #2), từ vị trí (7, 2) quân Mã có hai cách di chuyển để thực hiện đòn tấn công đôi.

1. Di chuyển tới vị trí (5, 3) sẽ tấn công được 3 quân của đối phương ở các vị trí (4, 1), (3, 2) và (4, 5).
2. Di chuyển tới vị trí (6, 4) sẽ tấn công được 2 quân của đối phương ở các vị trí (4, 5) và (7, 6).

Cho thông tin về vị trí quân Mã và vị trí của các quân đối phương, hãy tìm các vị trí mà quân Mã có thể di chuyển đến sau tối đa hai nước đi để thực hiện đòn tấn công đôi.

Lưu ý trong bài toán này không cần quan tâm quân đối phương là loại quân gì và chỉ cần tìm vị trí để quân Mã thực hiện đòn tấn công đôi, không cần quan tâm đến vị trí đó có được quân đối phương bảo vệ hay không.

[Input]

Dòng đầu tiên là số lượng test case **T** ($T \leq 50$). Thông tin về mỗi test case như sau:

Mỗi test case được cho trên 3 dòng, trong đó:

Dòng đầu tiên là số lượng quân đối thủ **N** ($2 \leq N \leq 16$).

Dòng thứ hai gồm hai số phân biệt với nhau bởi một dấu cách, mô tả vị trí (hàng, cột) của quân Mã.

Dòng thứ ba gồm **N** cặp số phân biệt với nhau bởi một dấu cách, mỗi cặp mô tả vị trí (hàng, cột) của quân đối thủ. Không có quân nào có vị trí trùng nhau hoặc trùng với quân Mã và không có quân nào đang bị tấn công bởi Mã.

[Output]

Đưa ra output trên **T** dòng tương ứng với **T** test case.

Mỗi test case in ra "#tc", với tc là số thứ tự của test case, đánh số bắt đầu từ 1, tiếp theo là một dấu cách và kết quả tương ứng của test case đó.

Kết quả in ra là số lượng vị trí mà quân Mã có thể di chuyển tới sau tối đa hai nước đi để thực hiện đòn tấn công đôi (số quân bị tấn công nhiều hơn hoặc bằng 2).

Xem output mẫu dưới đây.

Giải thích

#1 (Hình 3), Mã di chuyển đến vị trí (3, 3) để thực hiện đòn tấn công đôi, đây cũng là vị trí có thể tấn công đôi duy nhất sau tối đa hai nước di chuyển.

#2 (Hình 4), Có 2 vị trí Mã có thể tấn công đôi là (5,3) và (6,4).

#3 Có 1 vị trí duy nhất có thể thực hiện đòn tấn công đôi sau 2 nước di chuyển của quân Mã là (6, 2).

#4 Không có vị trí nào sau tối đa hai nước đi mà quân Mã có thể thực hiện đòn tấn công đôi

#5 Sau một nước đi, có 3 vị trí là: (2, 4), (4, 4), (5, 1) và sau hai nước đi có hai vị trí là: (5, 6), (6, 5) để quân mã có thể thực hiện đòn tấn công đôi.

Input:

5
2
4 5
1 2 2 1
4
7 2
3 2 4 1 4 5 7 6
8
6 4
2 6 4 8 8 2 2 2 4 1 8 6 8 8 1
8
2 4
5 8 1 1 8 1 3 4 5 7 3 1 8 2 8 8
9
3 2
8 4 3 6 4 3 7 7 4 8 1 6 6 3 6 5 1 2

Output:

#1 1

#2 2

#3 1

#4 0

#5 5