장기 포(包) 게임

C++

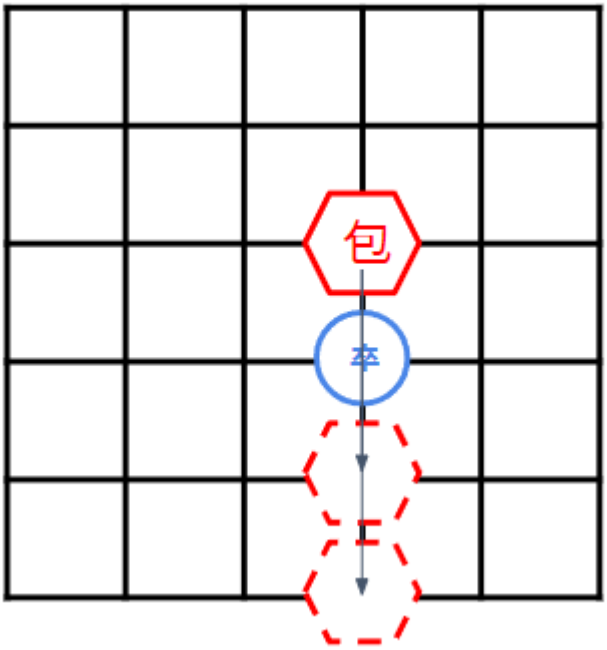
C++ 시간 제한	1,000 ms	메모리 제한	256 MB	정답률	3 / 6 (50.00%)

장기 게임에서의 포(包)는 아주 중요하고 강력한 말 중 하나이다.

아래는 포가 동작하는 방식에 대한 설명이다.

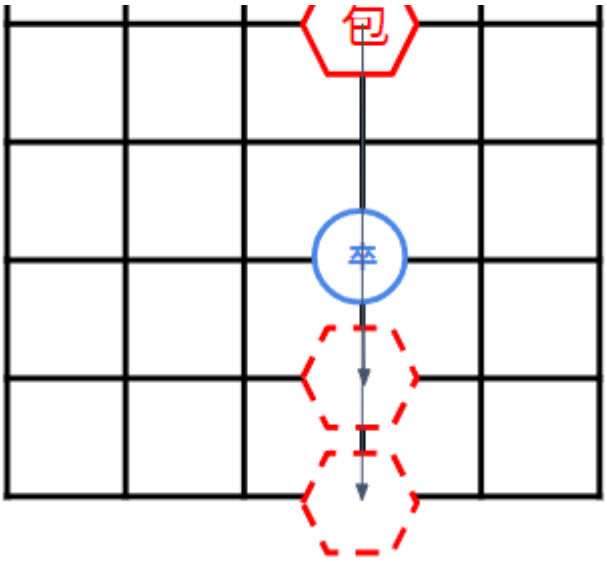
#1. 포는 다른 말을 하나 뛰어 넘어야만 이동이 가능하다.

아래의 예시에서는 (2, 3) 위치에 있는 포가 (3, 3)의 위치에 있는 쫄을 뛰어 넘어 (4, 3) 또는 (5, 3)의 위치로 이동이 가능하다.

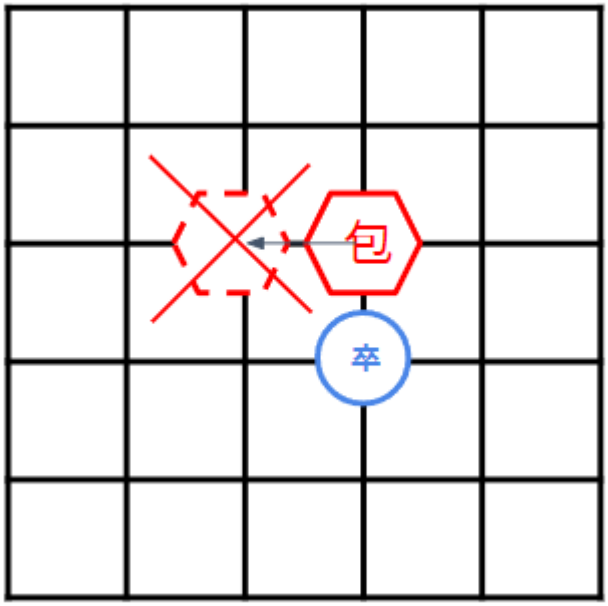


뛰어 넘을려고 하는 말이 바로 앞에 있지 않아도 이동이 가능하다.

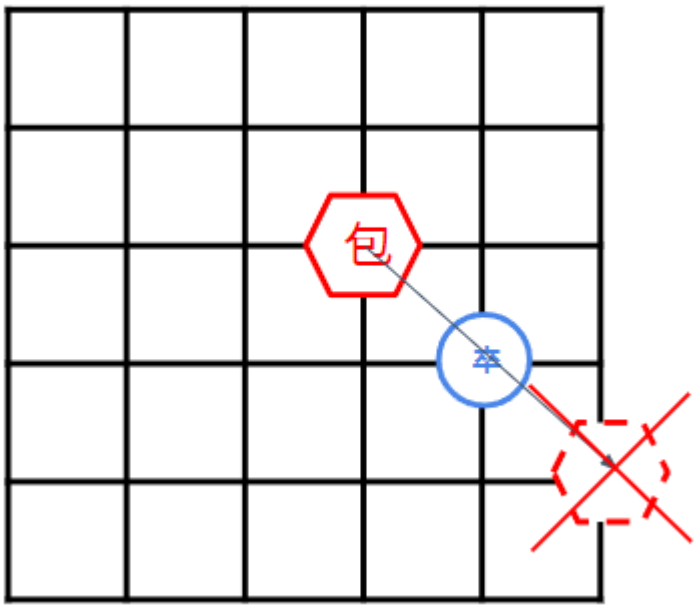
아래 예시에서는 (1, 3) 위치에 있는 포가 (3, 3) 위치의 쫄을 뛰어넘어 (4, 3) 또는 (5, 3)의 위치로 이동이 가능하다.



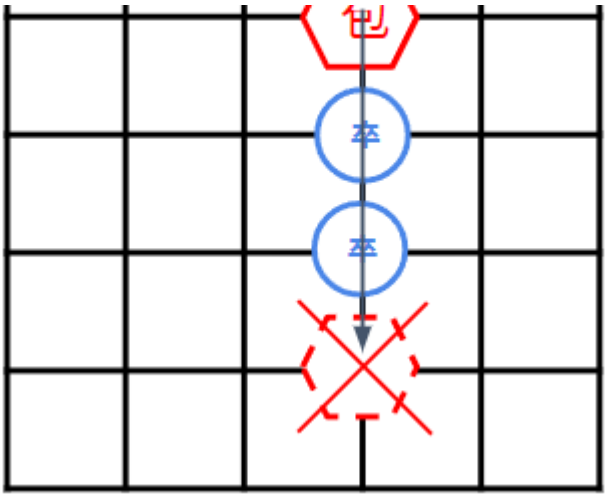
포는 다른 말을 뛰어넘지 않고서는 이동이 불가능하며,



상하좌우의 방향으로만 이동이 가능하며,



두개의 "붙어있는" 쫄을 뛰어 넘는것은 불가능하다.

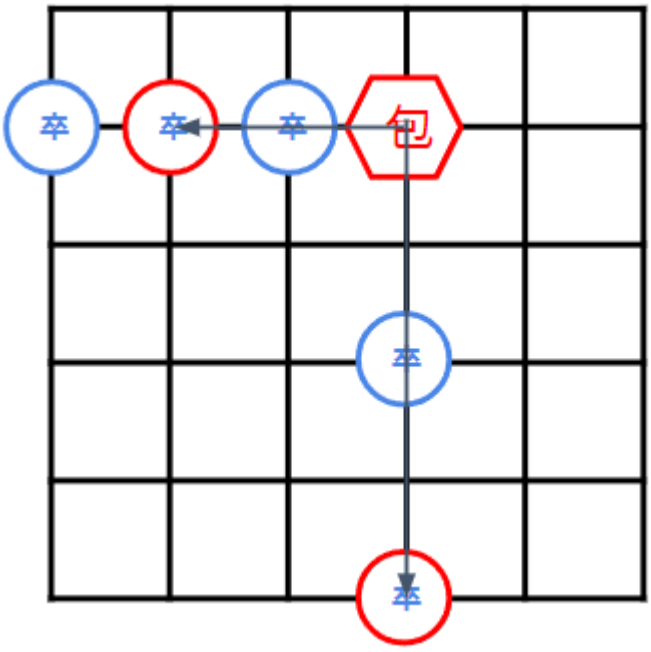


#2. 다른 하나의 말을 뛰어 넘었을 때, 해당 방향에 다른 말이 존재한다면, 해당 말을 먹을 수 있다.

아래와 같은 예시에서는 (1, 3) 위치의 포는 2개의 쫄을 먹을 수 있다.

- 왼쪽 방향으로 (1, 2) 위치의 쫄을 뛰어 넘어 (1, 1) 위치에 있는 쫄을 먹는다.
- 아래 방향으로 (3, 3) 위치의 쫄을 뛰어 넘어 (5, 3) 위치에 있는 쫄을 먹는다.

두 개의 연속으로 붙어있는 쫄을 뛰어넘는 것은 불가능하므로, (1, 0) 위치의 쫄은 먹을 수 없다.

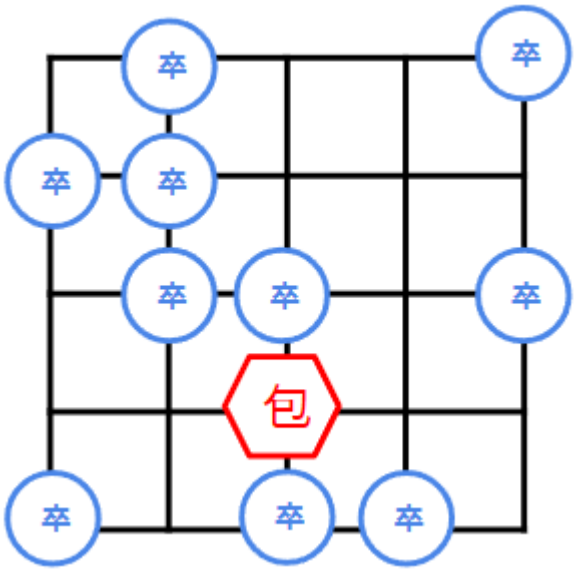


위 내용까지는 정상적인 장기말 포가 움직이는 방식이다.

여기서 포와 쫄만을 이용하여 새로운 **"장기 포 게임"을 만드려고 한다.**

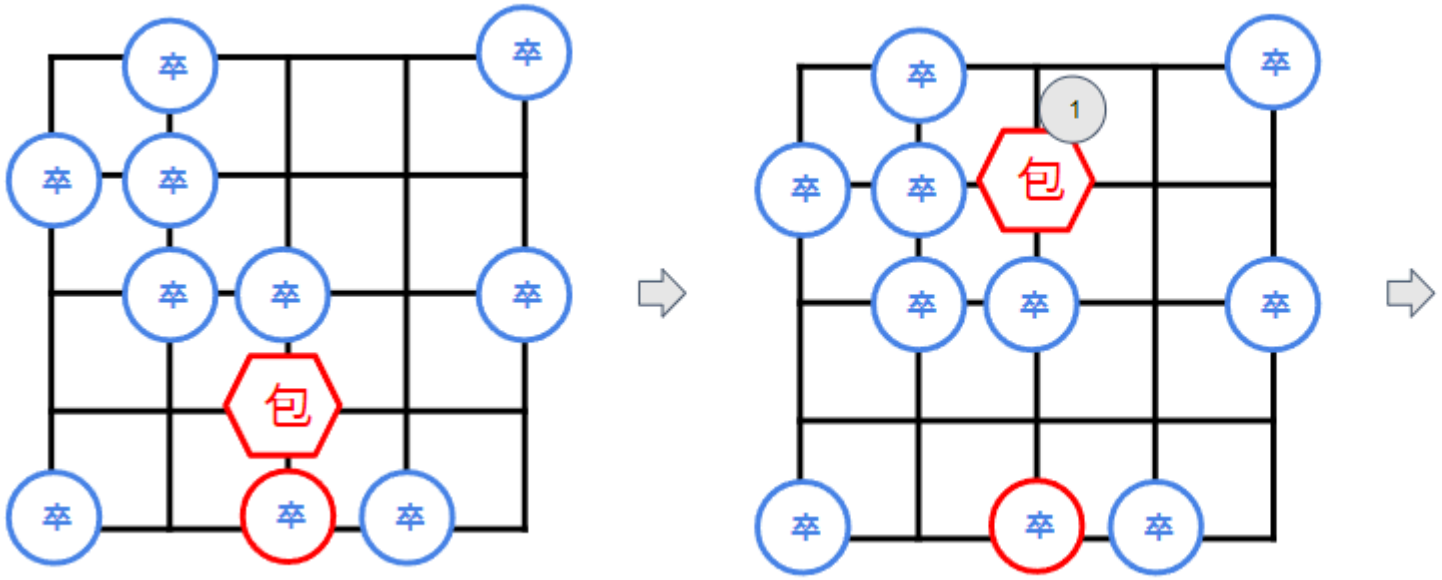
기존의 장기와 달라지는 규칙은 다음과 같다.

- 기존의 장기판이 아닌, N x N 크기의 장기판을 활용한다.
- 쫄은 여러개가 존재할 수 있고, 포는 단 하나만 존재한다.
- 포는 최대 3번까지 이동이 가능하다.

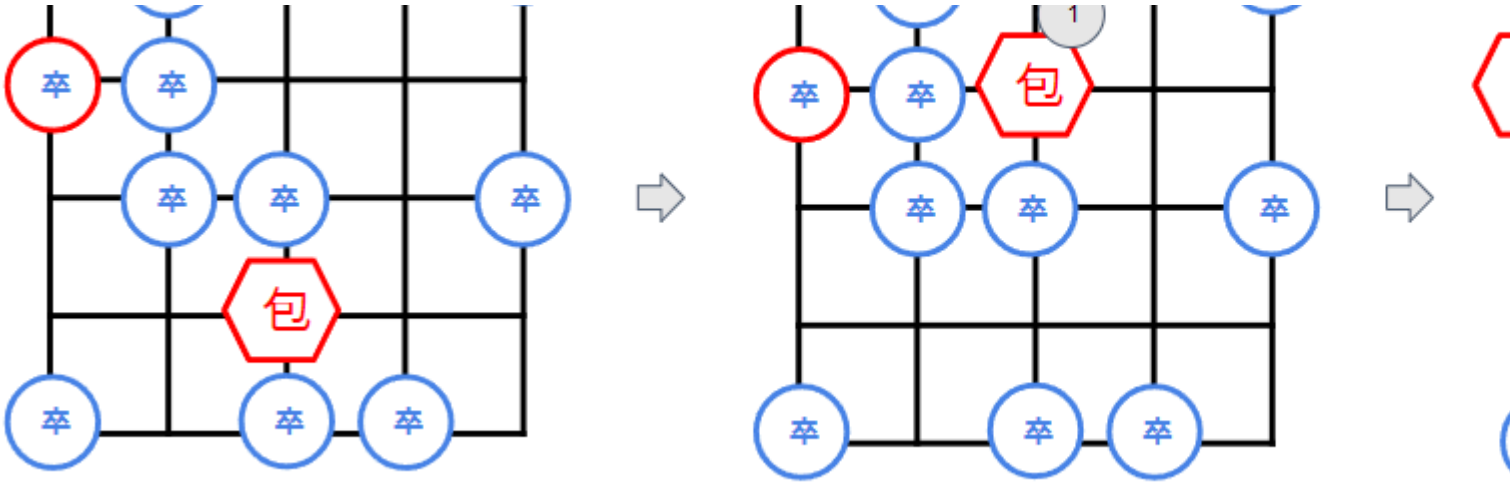


포는 3번까지 이동이 가능하니, 포가 먹을 수 있는 쫄들은 다음과 같다.

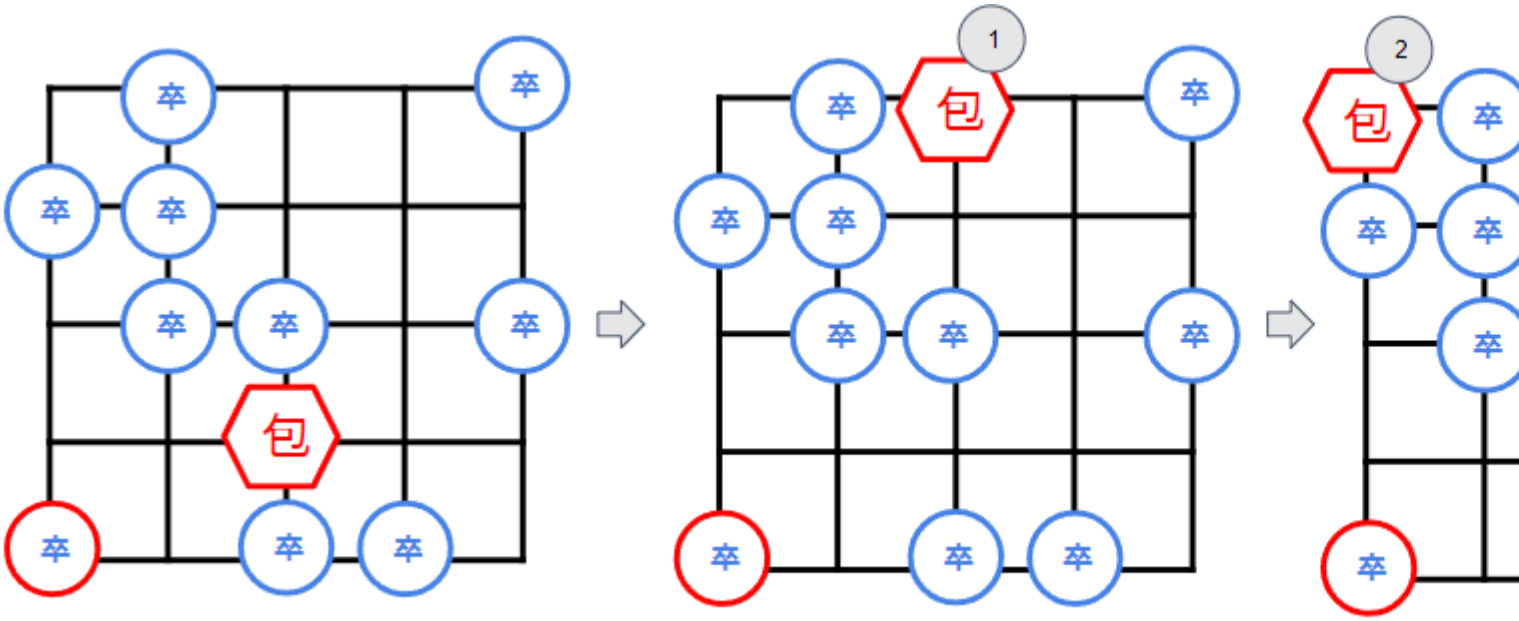
- **(4, 2) 위치의 쫄**



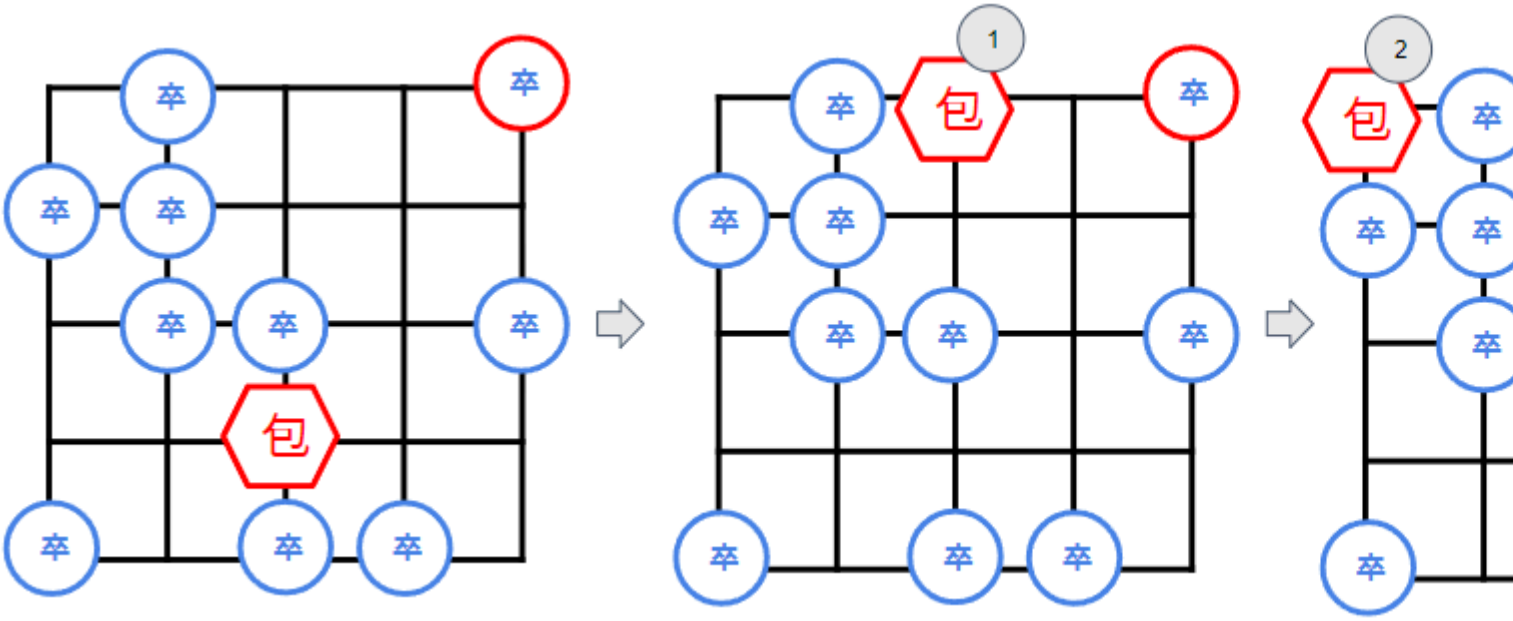
- **(1, 0) 위치의 쫄**



- **(4, 0) 위치의 쫄**



- **(0, 4) 위치의 쫄**



즉, 위의 장기 포 게임에서 포가 먹을 수 있는 **쫄의 개수는 총 4개** 이다.

[제한사항]

- 5 <= N <= 50
- 포는 2로 입력된다.
- 쫄은 1로 입력된다.
- 빈칸 (이동 가능한 칸) 은 0으로 입력된다.
- 포는 단 1개만 주어짐이 보장된다.
- 쫄의 개수는 최대 N * N - 1 개 까지 주어진다.

입력

첫번째 줄에 test case의 개수 T가 입력된다. (1 <= T <= 50)
두번째 줄부터 각 test case에 대한 입력이 주어진다.

각 test case의 첫번째 줄에는 N이 주어진다.
그리고 다음 줄부터 N개의 줄에 걸쳐 N x N 크기의 장기 포 게임의 게임판의 정보가 주어진다.

출력

각 test case 에 대하여 "#x" (x는 test case의 번호를 의미, 1부터 시작)를 출력하고, 하나의 공백을 둔 후 정답을 출력한다.

입력 예시 1

```
2
5
0 1 0 0 1
1 1 0 0 0
0 1 1 0 1
0 0 2 0 0
1 0 1 1 0
10
0 1 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 1 1 0 1 0 0 1 0 0
0 0 0 0 1 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0
0 0 0 0 2 0 1 0 1 0
0 0 0 0 0 0 1 0 0 0
1 0 1 1 0 0 1 0 0 0
```

출력 예시 1

```
#1 4
#2 15
```

Language: C++

```
1
```

HISTORY 제출하기