

BOJ 1890번 S2 점프

<https://www.acmicpc.net/problem/1890>

문제

$N \times N$ 게임판에 수가 적혀져 있다. 이 게임의 목표는 가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 점프를 해서 가는 것이다.

각 칸에 적혀있는 수는 현재 칸에서 갈 수 있는 거리를 의미한다. 반드시 오른쪽이나 아래쪽으로만 이동해야 한다. 0은 더 이상 진행을 막는 종착점이며, 항상 현재 칸에 적혀있는 수만큼 오른쪽이나 아래로 가야 한다. 한 번 점프를 할 때, 방향을 바꾸면 안 된다. 즉, 한 칸에서 오른쪽으로 점프를 하거나, 아래로 점프를 하는 두 경우만 존재한다.

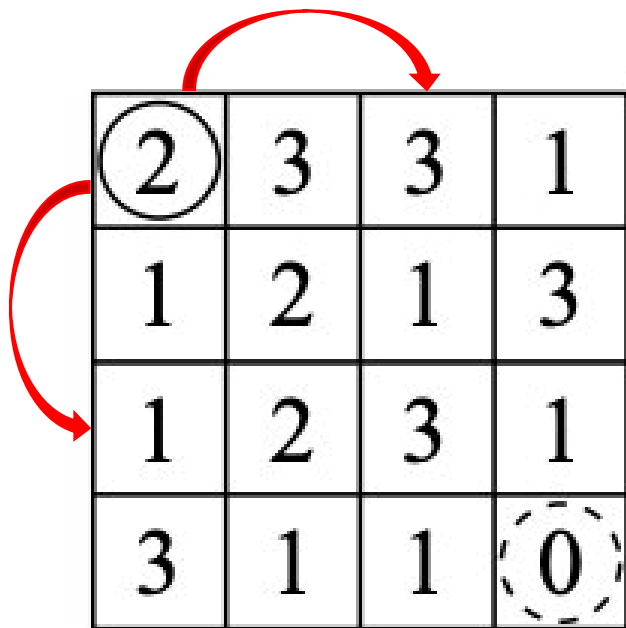
가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 이동할 수 있는 경로의 개수를 구하는 프로그램을 작성하시오.

입력

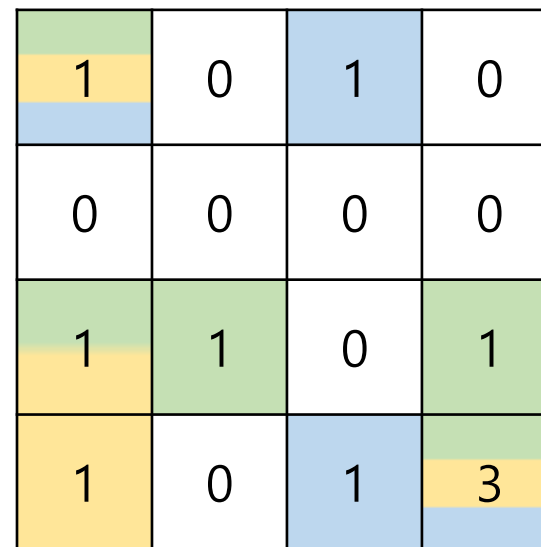
첫째 줄에 게임 판의 크기 N ($4 \leq N \leq 100$)이 주어진다. 그 다음 N 개 줄에는 각 칸에 적혀져 있는 수가 N 개씩 주어진다. 칸에 적혀있는 수는 0보다 크거나 같고, 9보다 작거나 같은 정수이며, 가장 오른쪽 아래 칸에는 항상 0이 주어진다.

출력

가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 문제의 규칙에 맞게 갈 수 있는 경로의 개수를 출력한다. 경로의 개수는 $2^{63}-1$ 보다 작거나 같다.



2	3	3	1
1	2	1	3
1	2	3	1
3	1	1	0

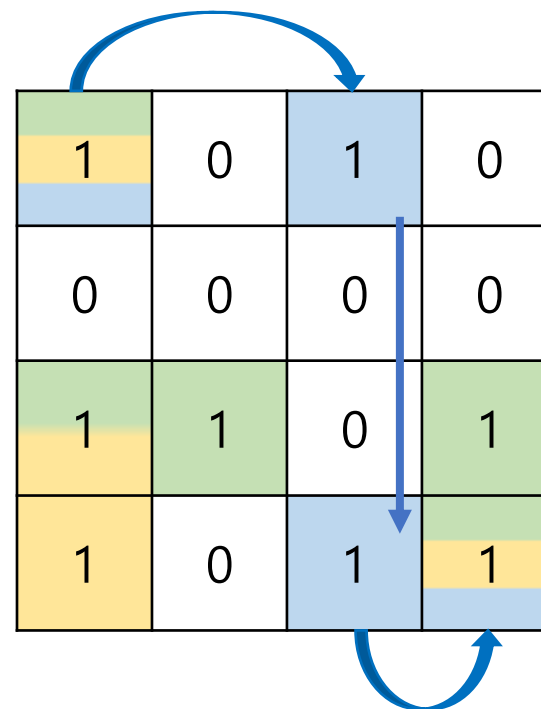
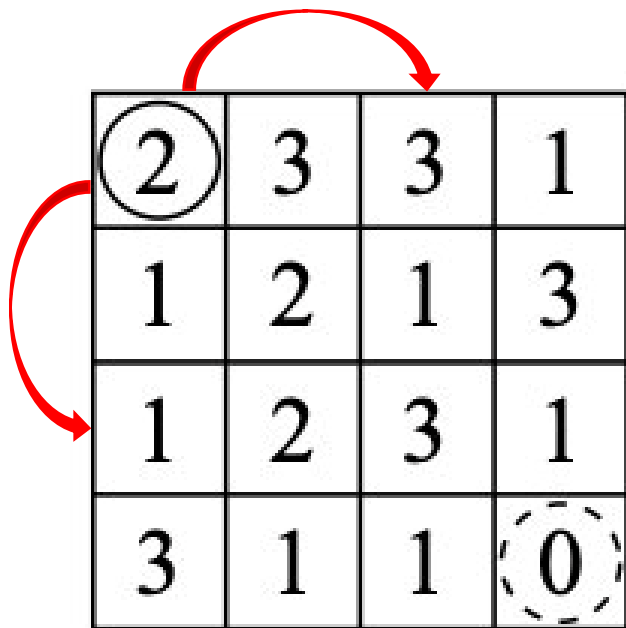


1	0	1	0
0	0	0	0
1	1	0	1
1	0	1	3

현재 칸에 오는 방법의 수 = 현재 칸에서 점프할 칸으로 가는 방법의 수

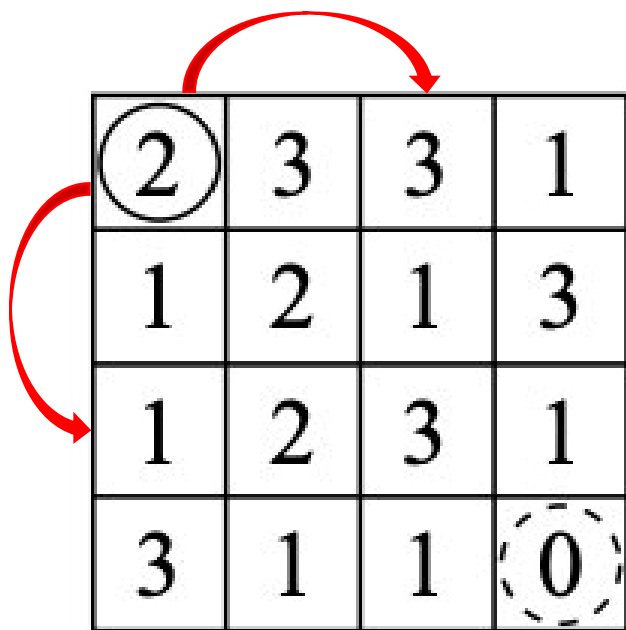
= (0, 0)에서 점프할 칸으로 가는 방법의 수

각 칸에 오는 방법의 수 = 각 칸으로 점프 가능한 칸으로 오는 방법의 수의 합

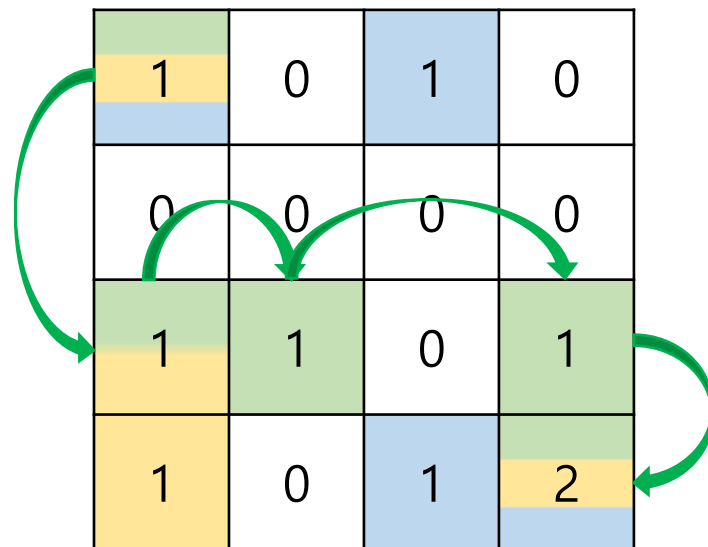


[풀이 방법]

현재 칸에서 갈 수 있는 칸의 방법의 수에
현재 칸까지 올 수 있는 방법의 수를 더한다



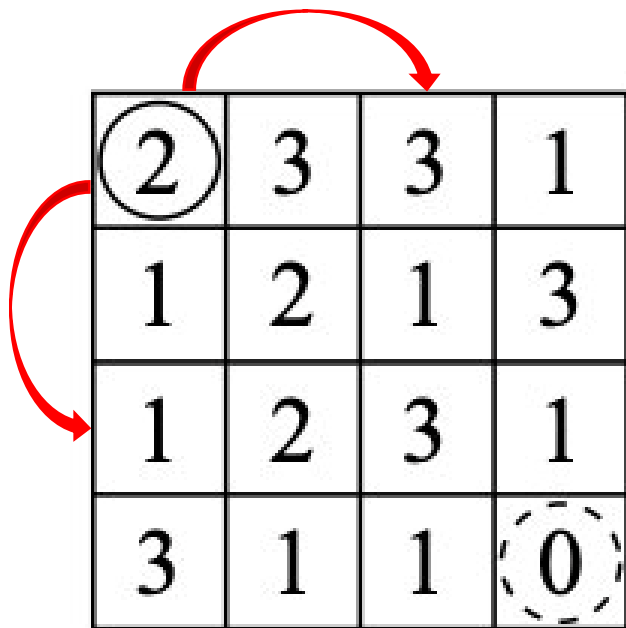
2	3	3	1
1	2	1	3
1	2	3	1
3	1	1	0



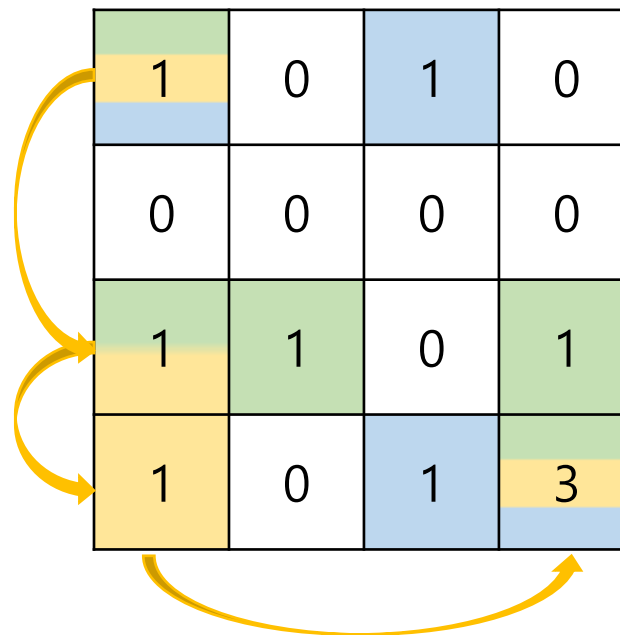
1	0	1	0
0	0	0	0
1	1	0	1
1	0	1	2

[풀이 방법]

현재 칸에서 갈 수 있는 칸의 방법의 수에
현재 칸까지 올 수 있는 방법의 수를 더한다



2	3	3	1
1	2	1	3
1	2	3	1
3	1	1	0



1	0	1	0
0	0	0	0
1	1	0	1
1	0	1	3

[풀이 방법]

현재 칸에서 갈 수 있는 칸의 방법의 수에
현재 칸까지 올 수 있는 방법의 수를 더한다

2	3	3	1
1	2	1	3
1	2	3	1
3	1	1	0

1	0	1	0
0	0	0	0
1	1	0	1
1	0	1	3

[초기화]

(0, 0) 칸에서 시작하므로 (0, 0) 값을 1로 초기화

[자료구조]

경로의 개수는 $2^{63}-1$ 보다 작거나 같으므로 long 배열 사용

[가지치기]

값이 0 이면 현재 칸까지 올 방법이 없으므로 계산 X

[주의사항]

마지막 칸을 계산하면 (값 $\times 2 \times 2$)가 되므로 계산하지 않고 반복문 종료