



Problem B
Color Walk

Alice and Bob are playing a game on a directed graph, fully visible to both of them. The graph has n nodes labelled 1 through n and directed arcs between nodes. Each arc is colored either red or black. In addition to the graph, there is a queue of capacity k, fully visible to both of them. At the start of the game, Alice places a game piece on node 1, and the queue is empty. Bob then inserts colors (either red or black) into the back of the queue until the queue is full. Now the main portion of the game begins: Alice removes the color at the front of the queue and moves the game piece along an outgoing arc of the same color from the current node to its destination node (which could be the same node in the case of a self-loop). The queue now has space for one more color, which Bob must now provide. Bob wins if Alice is ever unable to move the game piece, i.e., because there are no outgoing arcs of the appropriate color from the current node. Alice wins if she can keep playing forever (she really likes this game).

Given the complete description of the graph, determine the minimum capacity k for the queue such that no matter how well Bob plays, Alice can always win.

Input

Input begins with a line with a single integer t, $1 \leq t \leq 20$, denoting the number of test cases. Each test begins with a line with a single integer n, $1 \leq n \leq 12$, denoting the number of nodes in the graph. Next follow n lines each with n space-separated integers, where the jth number in the ith line is 1 if there is a red arc from node i to node j, and 0 otherwise. Next follow n lines each with n space-separated integers, where the jth number in the ith line is 1 if there is a black arc from node i to node j, and 0 otherwise. Remember that node 1 is always the start node.

Output

For each test case, print a single integer k on a single line equal to the minimum capacity for the queue such that no matter how well Bob plays, Alice can always win. If for all k, no matter how well Alice plays, Bob can always win, print 0.





Sample Input 1	Sample Output 1
3 3 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 1 2 0 1 1 0 0 0 0 1	2 1 0