D	Yet Another Longest Path Problem	Time Limit: 2 sec

We all know shortest path problem, right? When we have to move to one place from another using the shortest path. Now in this problem, we don't want shortest path but we are actually interested to find the longest path without exceeding some weight.

More specifically, you will be given a directed weighted graph with **N** nodes numbered **1** to **N**. Then you will be given a start node **U**, an end node **V** and a weight value **W**. You have to find the longest path from **U** to **V** where the cost of the path is maximum but not more than **W**.

Input

First line of the input will be one integer **N (1 ≤ N ≤ 100)**. Next **N** lines each will have **N** integers. The $\mathbf{j}^{th}$ integer of this $\mathbf{i}^{th}$ line, will be $\mathbf{w_{ij}}$ **(1 ≤** $\mathbf{w_{ij}}$ **≤ 100)**, weight of edge going to **j** from **i** or 0 if there is no road. $\mathbf{w_{ii}}$ will always be 0 that is will not be any self-loop.

In the next line of the input will be one integer **Q (1 ≤ Q ≤ 100)**, number of query. Next **Q** lines each will have three integers, **U V W (1 ≤ U, V ≤ N, 1 ≤ W ≤ 500)**.

Output

For each query, print the maximum cost of going **U** to **V** where the cost is not more than **W**. If it is not possible then print **-1**.

Sample I/O

Input	Output
5 0 2 4 0 0 3 0 0 0 0 9 2 0 2 0 0 0 0 0 1 0 6 0 0 0 5 3 1 5 3 1 6 3 1 11 1 3 39 1 3 3	5 5 10 39 -1