All-pairs shortest-paths in a digraph

Find shortest-path between every pair of vertices in a given directed graph (digraph). The digraph is given in the form of a weight matrix (aka cost adjacency matrix) and all-pairs shortest-paths of the graph is expected in the form of distance matrix where a value at row i and column j indicates the shortest distance from vertex i to vertex j.

Input

The input begins with the number t of test cases in a single line (t <= 100). Each test case begins with number of vertices n of the digraph in a new line (1 <= n <= 100) and the following n lines with the weight matrix of the graph. An entry at row i and column j in the weight matrix indicates the weight (aka cost) of the edge from vertex i to vertex j in the graph, or 32765 if there is no such edge.
Assumptions:
1. Weight of an edge is in the range [-32764, +32764].
2. Shortest path between any pair of vertices is in the range [-32764, +32764].
3. There are no cycles (a path starting and ending at a common vertex) of negative length.
4. Weight of a self loop (that is, the weight of an edge from vertex i to i) is always 0.

Output

For every test case print the distance matrix representing the shortest distance between each pair of vertices in the graph. An entry at row i and column j in the matrix should be the shortest distance from vertex i to vertex j in the graph if there is a path, otherwise 32765. A matrix of order n should be printed in n lines each line having n entries of a row of the distance matrix.

Example

Input:
4
2
0 32765
32765 0
2
0 1
2 0
4
0 32765 3 32765
2 0 32765 32765
32765 7 0 1
6 32765 32765 0
1
0

Output:
0 32765
32765 0
0 1
2 0
0 10 3 4
2 0 5 6

7 7 0 1
6 16 9 0
0