Shortest Paths in a Large Town

In a large town, there are n intersections connected by one-way roads. Each road has a certain travel time. The mayor wants to know the shortest time it takes to travel between any two intersections in the town.

You are given the number of intersections n and a list of m roads, each represented by three integers u, v, and w, meaning there is a one-way road from intersection u to intersection v with a travel time of w minutes.

Your task is to calculate the shortest travel time between all pairs of intersections. If there is no path between two intersections, the travel time should be considered as -1.

Input

The first line contains two integers n and m ($1 \leq n \leq 100,\ 0 \leq m \leq n^2$), representing the number of intersections and the number of roads, respectively.

The next m lines each contain three integers u, v, and w ($1 \leq u, v \leq n,\ 1 \leq w \leq 100$), describing a one-way road from intersection u to intersection v with a travel time of w minutes.

Output

Output n lines, each containing n integers. The j-th integer on the i-th line should represent the shortest travel time from intersection i to intersection j. If there is no path between intersections i and j, output -1 for that pair.

Sample

Input	Output
4 5 1 2 6 1 3 9 2 3 2 3 4 7 4 1 1	0 5 7 14 14 0 2 9 12 17 0 7 1 6 8 0

Notes

For the given example, we have 4 intersections and 5 roads. The output shows the shortest travel times between all pairs of intersections. For instance, the shortest time from intersection 1 to 4 is 14 minutes (1 → 2 → 3 → 4), while from 4 to 1 it's just 1 minute (direct road).