**Find Best Path**

You are given a directed graph with N*N* vertices (numbered 11 through N*N*) and M*M* weighted edges. For each vertex of the graph, find the minimum cost of a walk that passes through this vertex, or determine that the minimum does not exist, i.e. for any integer m*m*, there is a walk with cost smaller than m*m*.

**Notes:**

* A walk in a directed graph is a sequence of vertices v1,v2,…,vK*v*1​,*v*2​,…,*vK*​ for some positive integer K*K*, and an associated (possibly empty) sequence of edges e1,e2,…,eK−1*e*1​,*e*2​,…,*eK*−1​ such that for each valid i*i*, edge ei*ei*​ goes from vertex vi*vi*​ to vertex vi+1*vi*+1​.
* The cost of a walk is the sum of weights of edges e1,e2,…,eK−1*e*1​,*e*2​,…,*eK*−1​; if K=1*K*=1, the cost is 00.
* A walk *passes through* vertex u*u* if u*u* belongs to the sequence v1,v2,…,vK*v*1​,*v*2​,…,*vK*​.

**Input**

* The first line of the input contains a single integer T*T* denoting the number of test cases. The description of T*T* test cases follows.
* The first line of each test case contains two space-separated integers N*N* and M*M*.
* Each of the next M*M* lines contains three space-separated integers u*u*, v*v* and w*w* describing an edge from vertex u*u* to vertex v*v* with weight w*w*.

**Output**

For each test case, print N*N* lines. For each valid i*i*, if the minimum cost of a walk through vertex i*i* does not exist, the i*i*-th of these lines should contain the string "INF". Otherwise, it should contain a single integer — the minimum cost.

**Constraints**

* 1≤T≤501≤*T*≤50
* 1≤N,M≤1,0001≤*N*,*M*≤1,000
* ∣w∣≤106∣*w*∣≤106

**Subtasks**

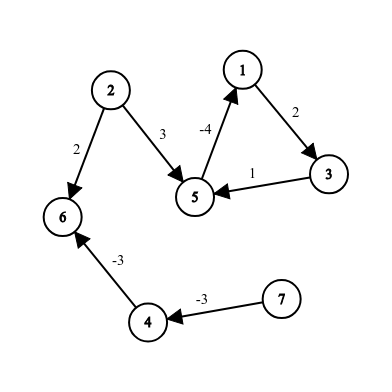
**Subtask #1 (30 points):** 1≤N,M≤1001≤*N*,*M*≤100

**Subtask #2 (70 points):** original constraints

**Sample 1:**

Input

Output

1

7 7

1 3 2

5 1 -4

2 5 3

2 6 2

3 5 1

4 6 -3

7 4 -3

INF

INF

INF

-6

INF

-6

-6