All Pairs Shortest Path

The Problem

Find the shortest path between every pair of nodes in a weighted directed graph.

The Input

The first line of input will be a number on a line by itself which is the number of test cases to run. For each test case, the first line will be two numbers separated by a space *N* and *E*, where *N* (1 <= *N* <= 1000) is the number of nodes in the graph and *E* (1 <= *E* <= 10000) is the number of edges. The graph nodes will be numbered 0 to *N*-1. Each of the next *E* lines contain three numbers *S, D* and *W* each separated by a space representing an edge in the graph from *S* (0 <= *S* < N) to *D* (0 <= *D* < N) with weight *W* (0 < *W* <= 1000).

The Output

For each test case, output the test case number on a line by itself. The following lines will contain the shortest paths matrix. That is the next n (0 <= n < N) lines will each contain a space separated list of the shortest path weight between node n and each other node in ascending node number order. If no path exists between a pair of node print 'NP'. See the example output below.

Sample Input

```
2
5 7
0 1 10
1 3 2
1 4 6
3 4 3
2 4 6
0 4 20
0 2 30
5 6
0 1 8
0 3 2
0 4 5
1 2 6
2 3 4
3 4 3
```

Sample Output

```
1
0 10 30 12 15
NP 0 NP 2 5
NP NP 0 NP 6
NP NP NP 0 3
NP NP NP NP 0
2
0 8 14 2 5
NP 0 6 10 13
NP NP 0 4 7
NP NP NP 0 3
NP NP NP NP 0
```