Johnsons Algorithm

Johnson's algorithm solves the all-pairs shortest path problem in a weighted, directed graph.

Input

t [number of test graphs]

[description of each graph]
n [number of nodes, 1<=*n*<=100]
m [number of edges, *m*>=0]
[next the list of edges, one edge per line]
u v w [*e(u,v)* - edge from node *u* to node *v* with weight *w*]
    [1<=*u*<=n, 1<=*v*<=n, -1000<=*w*<=1000]
...   [next edge]

[next graph]
...

Output

If the *i*-th test graph has negative weight cycles, then the answer should be:

graph i no [where '*i*' is the number of the graph, 1<=*i*<=*t*]

Otherwise you should output the following data:

graph i yes

[vector of function h(v)]
h_1 h_2 ... h_{n+1}

[matrix *d[u,v]*, the solution of the all-pairs shortest path problem]
$d_{1,1}$ $d_{1,2}$ ... $d_{1,n}$
$d_{2,1}$ $d_{2,2}$ ... $d_{2,n}$
... ...
$d_{n,1}$ $d_{n,2}$ ... $d_{n,n}$
[if the path doesn't exist, you should output # instead]

Example

Input:
6

2
2
1 2 -2
2 1 1

6
8
1 2 8
1 6 6
6 2 3
2 3 -1

3 6 -2
6 5 -2
5 4 2
3 4 3

4
4
1 2 1
2 3 2
3 4 3
4 1 0

2
0

1
0

2
2
1 2 -1
2 1 0

Output:
graph 1 no

graph 2 yes

0 0 -1 -3 -5 -3 0

0 8 7 5 3 5
0 -1 -3 -5 -3
1 0 -2 -4 -2
0 #
2 0
3 2 0 -2 0

graph 3 yes

0 0 0 0 0

0 1 3 6
5 0 2 5
3 4 0 3
0 1 3 0

graph 4 yes

0 0 0

0 #
0

graph 5 yes

0 0

0

graph 6 no