

### 3. All Pairs Shortest Paths

임의의 한 점에서 임의의 다른 점으로 가는 경로가 항상 존재  
최단거리의 총합을 그대로 출력하면 된다.

Negative Cycle이 없는 Weighted **Directed** Graph  $G(V, E)$ 가 주어질 때. 모든 정점 쌍 사이의 최단 경로를 구하는 알고리즘을 구현하라. 그래프는 ~~인접행렬 형식으로 주어진다.~~ 정점이 총  $N$ 개라면  $N \times N$  행렬을 통해 각 정점에서 다른 정점으로 가는 간선의 가중치가 주어진다. 가중치는 양 또는 음의 정수로 주어진다. 이를 통해 모든 정점 쌍 사이의 **최단 경로 행렬**을 구하고, 이를 모두 더한 값을  $1$ 억으로 나눈 나머지를 구하시오. 알고리즘은 최대한 효율적으로 작성하라. 채점을 위한 컴파일 시에 최적화 옵션은 쓰지 않는다.

#### [제약사항]

정점의 개수  $1 \leq N \leq 200$ , 간선의 개수  $1 \leq E \leq 10,000$

\* 수행 시간이 다른 학생들에 비해 지나치게 길면 감점 요인이 됨.

#### [입력]

입력 파일에는 10 개의 테스트 케이스가 주어진다. 각 케이스는 2 줄로 이루어진다. 첫 줄에는 **정점의 개수  $N$** 이 주어지고 공백을 두고 **간선의 개수  $E$** 가 주어진다. 다음 줄에는  $E$ 개의 간선 정보가 공백을 두고 주어지는데, 각각의 간선 정보는 각각 **출발 정점, 도착 정점, 가중치**로 이루어진 3개의 값이 공백을 두고 주어진다.

이 때, **정점의 번호는 1부터  $N$ 까지**로 매긴다. **간선의 가중치는 0을 제외한 -1000부터 +1000**까지 주어진다. 입력파일의 이름은 "input1.txt"이다.

#### [출력]

각 테스트 케이스에 대해서, 케이스의 번호를 "#x" 의 형식으로 출력한 후(여기서  $x$  는 테스트 케이스 번호), 공백을 하나 둔 다음 주어진 케이스에서 각각 **최단 경로 행렬을 모두 더한 값을 1억으로 나눈 나머지를** 구하시오. 출력 결과물을 "output1.txt"로 저장한다.

#### [예제]

입력 (input1.txt)

2 2	← 1번 케이스
1 2 100 2 1 -50	
3 3	← 2번 케이스
1 2 100 2 3 -50 3 1 30	
...	

**출력 (output1.txt)**

#1 50
#2 240
...