

## Problem I. Traffic flow

Source file name: I.c, I.cpp, I.java  
Input: Standard  
Output: Standard  
Author(s):

A city has  $n$  intersections and  $m$  bidirectional roads connecting pairs of intersections. Each road has a certain traffic flow capacity, measured in cars per minute. There is a path from every intersection to every other intersection along some sequence of roads. The road maintenance department is over budget and needs to close as many roads as possible without disconnecting any intersections. They want to do it in such a way that the minimum capacity among all of the remaining roads is as large as possible.

### Input

The first line of input gives the number of cases,  $T$ .  $T$  test cases follow. Each one starts with a line containing  $n$  ( $0 \leq n \leq 100$ ) and  $m$  ( $0 \leq m \leq 10000$ ). The next  $m$  lines will describe the  $m$  roads, each one using 3 integers,  $u$ ,  $v$  and  $c$  ( $0 \leq u, v \leq n$ ), ( $0 \leq c \leq 1000$ ).  $u$  and  $v$  are the endpoints of the road and  $c$  is its capacity.

### Output

For each test case, output one line containing "Case #x: " followed by the capacity of the minimum-capacity remaining road.

### Example

Input	Output
2 2 3 0 1 10 0 1 20 0 0 30 4 5 0 1 1 3 1 2 1 2 3 2 3 4 0 2 5	Case #1: 20 Case #2: 3