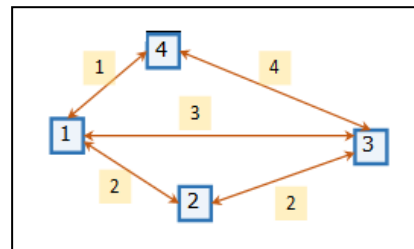




Задача В. Аварийна група

Електроразпределителното дружество „BEZ“ решава да подмени аварийните автомобили с електромобили. Обектите, за които отговаря аварийната група, в която работи Петър, са разположени в няколко съседни селища. На картата на района са означени селищата, пътищата, които ги свързват и дължините на пътищата. Между всяка двойка населени места съществува най-много един двупосочен път. Няма пътища, водещи от населено място директно към самото него. След като се получи сигнал за авария, Петър трябва да състави списък с различните маршрути до селището на аварията, дължината на които не надвишава разстоянието, което електромобилът може да измине без презареждане. Маршрутът трябва да е само за отиване до мястото на аварията и да е такъв, че да минават през дадено населено място не повече от веднъж. Напишете програма, която изготвя желаните списък от маршрути, дължините на които не превишават максималния пробег на електромобила.



Вход: При едно изпълнение, програмата трябва да обработва няколко тестови примера. За всеки тестов пример, от първия ред на стандартния вход се въвеждат две цели числа, разделени с интервал – броят V на населените места в района на аварийната група, номерирани от 1 до V и броят R на пътищата между двойки населени места ($2 \leq V \leq 64$, $2 \leq R \leq 64$). От следващите R реда се въвеждат по три цели числа C_1 , C_2 и D . C_1 и C_2 са номерата на две населени места, свързани с път, а D – разстоянието между тях ($1 \leq D \leq 9999$). От последния ред на теста се въвеждат три цели числа: номерът S на населеното място, от което тръгва аварийната група, номерът T на населеното място, в което е възникнала аварията и дължината M на пътя, който електромобилът може да измине без презареждане.

Изход: За всеки тест, на отделни редове на стандартния изход, програмата трябва да изведе намерените маршрути, подредени в нарастващ ред на дължините им, а при равни дължини – в лексикографски нарастващ ред. Редът започва с дължината на маршрута, следвана от ‘:’ и интервал. Следват номерата на населените места, които формират маршрута, разделени с интервали. Ако няма намерени маршрути, които удовлетворяват условията, програмата трябва да изведе ред с думата **No**.

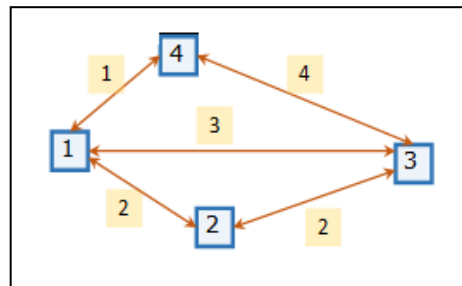
Пример:

Вход	Изход
4 5	3: 1 3
1 2 2	4: 1 2 3
1 3 3	1: 1 4
1 4 1	7: 1 3 4
2 3 2	8: 1 2 3 4
3 4 4	
1 3 4	
4 5	
1 2 2	
1 3 3	
1 4 1	
2 3 2	
3 4 4	
1 4 10	



Task B. Emergency group

Electricity supplying company „BEZ“ decided to replace the emergency group cars with electro mobiles. The objects for which is responsible the group of Peter are the villages of one region. The map of the region shows the villages, roads that connect them, and the lengths of the roads. There is no more than one road between a pair of villages and no road that starts and ends in the same village. After emergency call is received, Peter has to create a list of the different paths from the base of the emergency group to the village that send the emergency call, which are no longer than the distance the electro mobile could pass without recharging. The path has to be only from the base to the village with a problem and have not to pass through a village more than once. Write a program to prepare the asked list of paths.



Input: The program has to be able to solve more than one test case per run. For each test case the first line of the standard input will contain two integers separated by interval: the number V of the villages of the region, labeled from 1 to V and the number R of the roads between some couples of villages ($2 \leq V \leq 64$, $2 \leq R \leq 64$). Each of the next R rows will contain three integers C_1 , C_2 and D . C_1 and C_2 are the labels of two villages connected by a road and D – the length of the road ($1 \leq D \leq 9999$). The last line of the test case will contain three integers too: the label S of the village where the base of the group is placed, the label T of the village that send an emergency call and the maximal distance M that the electro mobile could pass without recharging.

Output: For each test case the program has to print on separate lines the found paths, sorted in increasing order of their lengths, and when some paths have the same length – in increasing lexicographical order. The line has to start with the length of the path, the sign ‘:’ and one interval. Then the labels of the villages follow in order of visiting, separated by single intervals. If there is no appropriate paths for the test case then the program has to print a line with the word **No**.

Example

Input	Output
4 5	3: 1 3
1 2 2	4: 1 2 3
1 3 3	1: 1 4
1 4 1	7: 1 3 4
2 3 2	8: 1 2 3 4
3 4 4	
1 3 4	
4 5	
1 2 2	
1 3 3	
1 4 1	
2 3 2	
3 4 4	
1 4 10	