Jiajia and WinD quarreled. They got angry with each other. They went far from each other. They regretted. They missed each other so much. They wanna say sorry. They wanna meet in a city as soon as possible. They wanna be together again and forever.

Each of their routes should be simple - no city should be reached more than once. Their routes should be disjoint - no city should be reached by them both, except the last one - their meeting city. They don't want anyone to travel more, so the distance covered by them should be exactly the same.

Input

The input consists of at most 20 test cases. Each case begins with a line containing four integers n, m, J, W ($0 < n < 201, 0 < m < 3n/2, 1 \leq J, W \leq n$), the number of cities and bidirectional roads in the city network, and the initial city number of Jiajia and WinD. Each of the following m lines describes a road. There are three integers in each road description: i, j, d, indicating there is a bidirectional road connected i and j, with length d ($1 \leq i, j \leq n, 1 \leq d \leq 10$) Each pair of cities can be connected by at most one road. It is assumed that every pair of cities can be reached from each other. J does not equal to W. The last case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the distance covered by each person in the first line. The next two lines should contain the route plan for Jiajia and WinD, respectively. Each route plan begins with an integer c, the number of cities visited. The next c integers form the list of cities visited. There is always a solution.

Sample Input

```
4 4 1 4
1 2 1
2 3 1
3 4 2
1 3 1
4 5 1 4
1 2 2
1 3 5
2 3 4
2 4 1
3 4 1
0
```

Sample Output

```
Case 1: 2
3 1 2 3
2 4 3

Case 2: 5
2 1 3
3 4 2 3
```