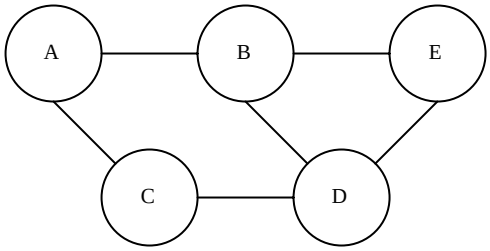
Problem 2: Bandwidth

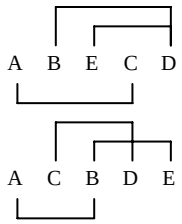
Given an undirected graph (*V*, *E*), where *V* is a set of vertices and *E* is a set of edges, and an ordering on the elements in V, then the *bandwidth* of a vertex *v* is defined as the maximum distance in the ordering between *v* and any node to which it is directly connected in the graph, or 0 if v is not connected to any other vertices. The bandwidth of an ordering is defined as the maximum of the bandwidths of the vertices in the ordering.

The distance between two vertices A and B in an ordering is just one more than the number of vertices between A and B in the ordering. So if an ordering includes … A P Q R B …, then the distance between A and B in that ordering is 4 (assuming A is directly connected to B in the graph).

Example

For example, consider the graph shown below on the left.





The vertices in this graph may be ordered in many different ways, two of which are illustrated on the right. For the first ordering (A, B, E, C, D), the bandwidths of the vertices A, B, E, C and D are 3, 3, 2, 3, and 3; the bandwidth of the ordering is thus 3, the maximum vertex bandwidth. The second ordering (A, C, B, D, E) has vertex bandwidths of 2, 2, 2, 2, and 2, which yields an ordering bandwidth of 2.

The Problem

Your task is to write a program that will determine the ordering of the vertices in a graph that will yield the *minimum* bandwidth.

Input

There will be multiple input cases to consider. For each case there will be a single input line containing a description of a graph. A line will consist of a series of vertex descriptions separated from each other by semicolons. Each vertex description will consist of a vertex name (a single uppercase letter in the range 'A' to 'Z') followed by a colon and the name of one (or more) of the vertices to which it is directly connected. The input line for the last case will be followed by a line containing only $ in the first column. The graph will contain no more than 10 vertices.

Output

For each input case, display the case number (1, 2, …), the names of the vertices (separated from each other by a space), and the minimal bandwidth (in parentheses). If more than one ordering produces the same minimal bandwidth, then display the one that has the smallest lexicographic ordering. Display a blank line after the output for each case.

Sample Input	Output for the Sample Input
A:BC;B:DE;E:D;C:D A:DW;B:AES;S:TD;D:E;S:E;T:W $	Case 1: A C B D E (2) Case 2: B E A S D W T (3)