3175 The Take-Grant Protection Model

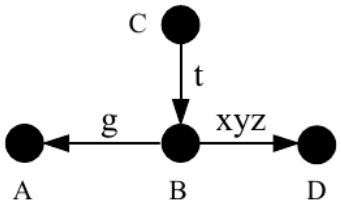
In this problem we consider a limited form of the "take-grant protection model." In this model, a system is represented as a graph. Vertices are normally either subjects (users or processes, the active agents in the system) or objects (for example, files). Directed edges between vertices are labeled to indicate the rights that the source vertex has with respect to the destination vertex.

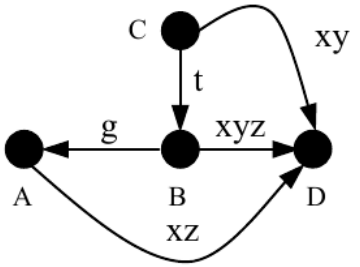
In this limited form of the model, we will consider only subject vertices.

Two rights of particular interest give this model its name. The "take" right, indicated by the letter '`t`', means a subject can take (a copy of, as its own) a right possessed by the vertex pointed to by the edge labeled "t". The "grant" right, identified by the letter '`g`', means a subject can give (a copy of) a right relative to a vertex to another subject. When the "take" or "grant" occurs, a new edge is created in the graph if necessary, or the additional rights are added to the label on an existing edge. Note that rights are always specified with respect to a particular vertex.

As an example, consider the graph shown on the left below. Subject B possesses rights `x`, `y` and `z` to D as indicated by the edge from B to D labeled "xyz". Likewise, B has the right to grant to A some or all of the rights it has; this is indicated by the edge from B to A labeled "g". Finally, subject C has the right to take (copy) from B any of the rights it possesses (indicated by the edge from C to B labeled "t").

Suppose B should grant the rights "`xz` to D" to subject A, and C should take the rights "`xy` to D" from B. Then the resulting graph would be as shown on the right. Note that C could also have taken the "grant to A" right possessed by B, which would have added an edge from C to A labeled "g".





Rights can be exercised at any time, and the same right can be exercised repeatedly. Subjects may also perform two additional actions. "Create" can be used by a subject S to add a new vertex V to the graph and an edge from S to V; this new edge can be labeled with any rights desired. "Remove" allows a subject S to remove some or all of the rights it possesses to a subject V; if all of the rights from S to V are removed, so is the edge from S to V. If a vertex has no edges to or from it, then it is removed. Every subject always has the right to create a new subject vertex with arbitrary rights to it, and to remove any rights it already possesses to vertices.

Given the graph for an initial system state, and a graph showing a possible future state, we wish to know whether that future system state could be reached using only the four types of actions (take, grant, create, and remove) just described. Rights are represented by a set of 26 or fewer lowercase letters (with '`g`' and '`t`' indicating "grant" and "take" rights). The subject vertices in the possible future state are the same as those in the initial state (which does not preclude the possibility that additional subject vertices may be created and removed between the initial and possible future state).

Input

There will be multiple cases to consider.

The input for each case consists of the graph describing the initial system state, followed by the graph for the possible future system state. Each of these graphs is specified by several input lines. The first line contains an integer NV giving the number of vertices (never more than 10) and an integer NE giving the number of edges. The labels for the vertices are the first NV uppercase letters (`A`, `B`, ...). Then there is a sequence of NE lines, one for each edge. Each of these lines begins with a pair of uppercase letters S and D (separated by a space) identifying the source and destination vertices for an edge in the graph, a space, and a string containing between 1 and 26 lowercase alphabetic characters identifying the rights S has with respect to D.

The last case is followed by a line containing two integer zeroes.

Output

For each case, display the case number (1, 2, ...) and the word '`yes`' or '`no`' to indicate if the second graph could, or could not be obtained from the first graph by some sequence of take, grant, create, and remove rule applications.

Separate the output for consecutive cases with a blank line.

Sample Input

```
4 3
B A g
C B t
B D xyz
4 5
B A g
C B t
B D xyz
C D xyz
A D xyz
3 2
A B x
C B ty
3 2
A B x
C B tyx
0 0
```

Sample Output

```
Case 1: yes

Case 2: no
```