In a graph G, *contraction* of an edge e with endpoints u, v is the replacement of u and v with a single vertex such that edges incident to the new vertex are the edges other than e that were incident with u or v. The resulting graph has one less edge than G. A graph H is a *minor* of a graph G if a copy of H can be obtained from G via repeated edge deletion, edge contraction and isolated node deletion.

Minors play an important role in graph theory. For example, every non-planar graph contains either the graph $\mathbf{K}_{3,3}$ (i.e., the complete bipartite graph on two sets of three vertices) or the complete graph $\mathbf{K}_5$ as a graph minor.

Write a program to find a graph minor $\mathbf{K}_{n,m}$ or $\mathbf{K}_n$ in an ***undirected connected simple graph***.

Input

The input consists of several test cases. The first line of each case contains an integers V ($3 \le V \le 12$), the number of vertices in the graph, followed by a string in format '`K`n' or '`K`n`,`m' ($1 \le n, m \le V$), the graph minor you're finding. The following V lines contain the adjacency matrix of the graph ('`1`' means directly connected, `0`' means not directly connected).

The diagonal elements of the matrix will always be '`0`', and the element in row i column j is always equal to the element in row j column i. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the string '`Found`' or '`Not found`'.

Sample Input

```
5 K2,2
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
4 K3
0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0
4 K2,2
0 1 0 1
1 0 1 1
0 1 0 1
1 1 1 0
5 K2,2
0 1 0 0 1
1 0 0 0 1
0 0 0 1 1
0 0 1 0 1
1 1 1 1 0
5 K4
0 1 0 1 1
1 0 1 1 0
0 1 0 1 1
1 1 1 0 1
1 0 1 1 0
0
```

Sample Output

```
Case 1: Not found
Case 2: Found
Case 3: Found
Case 4: Not found
Case 5: Found
```