The Center City fire department collaborates with the transportation department to maintain maps of the city which reflects the current status of the city streets. On any given day, several streets are closed for repairs or construction. Firefighters need to be able to select routes from the firestations to fires that do not use closed streets.

Central City is divided into non-overlapping fire districts, each containing a single firestation. When a fire is reported, a central dispatcher alerts the firestation of the district where the fire is located and gives a list of possible routes from the firestation to the fire. You must write a program that the central dispatcher can use to generate routes from the district firestations to the fires.

Input

The city has a separate map for each fire district. Streetcorners of each map are identified by positive integers less than 21, with the firestation always on corner #1. The input file contains several test cases representing different fires in different districts.

- The first line of a test case consists of a single integer which is the number of the streetcorner closest to the fire.
- The next several lines consist of pairs of positive integers separated by blanks which are the adjacent streetcorners of open streets. (For example, if the pair 4 7 is on a line in the file, then the street between streetcorners 4 and 7 is open. There are no other streetcorners between 4 and 7 on that section of the street.)
- The final line of each test case consists of a pair of `0`'s.

Output

For each test case, your output must identify the case by number (‘`CASE 1:`’, ‘`CASE 2:`’, etc). It must list each route on a separate line, with the streetcorners written in the order in which they appear on the route. And it must give the total number routes from firestation to the fire. **Include only routes which do not pass through any streetcorner more than once.** (For obvious reasons, the fire department doesn’t want its trucks driving around in circles.)

Output from separate cases must appear on separate lines.

Sample Input

```
6
1 2
1 3
3 4
3 5
4 6
5 6
2 3
2 4
0 0
4
2 3
3 4
5 1
1 6
7 8
8 9
2 5
5 7
3 1
1 8
4 6
6 9
0 0
```

Sample Output

```
CASE 1:
   1   2   3   4   6
   1   2   3   5   6
   1   2   4   3   5   6
   1   2   4   6
   1   3   2   4   6
   1   3   4   6
   1   3   5   6
There are 7 routes from the firestation to streetcorner 6.
CASE 2:
   1   3   2   5   7   8   9   6   4
   1   3   4
   1   5   2   3   4
   1   5   7   8   9   6   4
   1   6   4
   1   6   9   8   7   5   2   3   4
   1   8   7   5   2   3   4
   1   8   9   6   4
There are 8 routes from the firestation to streetcorner 4.
```