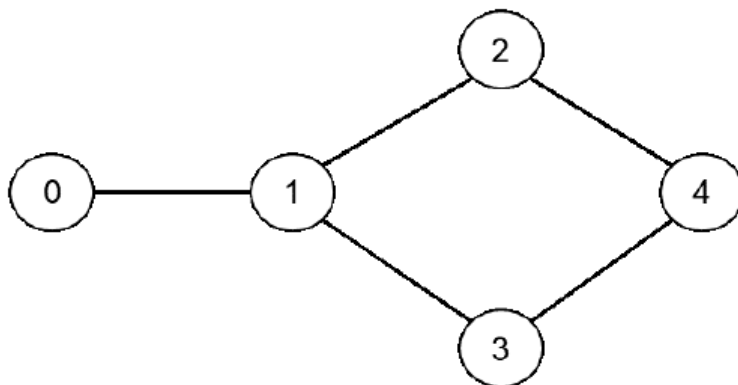


8266 Network Report

For time efficiency, data transmission is often done along the shortest path (with minimum number of hops) between network nodes. In the network graph (undirected and unweighted) below, the length of the shortest path from node 0 to node 2 is 2, and the length of the shortest path from node 0 to node 4 is 3. For a network graph with n nodes, we want to find all-pairs shortest paths, and provide a summary report on how many shortest paths are there for each length (from 1 to the maximal length). That is, this summary report should contain the total number of shortest paths of length 1, the total number of shortest paths of length 2, and so on and so forth.

Note that if multiple shortest paths from node i to node j ($i \neq j$) exist, they should collectively count as one. For example, though there are two shortest paths from node 0 to node 4, only one path is counted.

Please write a program to output the above summary report from an input connected graph.



Technical Specification

1. $2 \leq n \leq 200$, nodes are numbered as $0, 1, 2, \dots, n-1$.
2. Any counting number is within the range of integer.

Input

The input contains multiple instances of an undirected and unweighted connected graph (not multi-graph).

For each instance, the first line contains a single integer n . The second line contains a single integer e , denoting the number of edges in the graph. For the next e lines, each line contains two integers i, j , denoting that there is an edge connecting node i and node j . There is a '0' on a single line following the last test instance. There are at most 10 test instances.

Output

For each instance, there are several lines of output pending on the input. Each line of output consists of two integer numbers separated by a single space, where the first number is the path length and the second number is the total number of shortest paths of the corresponding length. The output should display the results from length 1 to the maximal length in the graph incrementally.

Sample Input

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

Sample Output

```
1 6
1 10
2 8
3 2
```