Shortest Cycle Hack

Time: 1 sec / Memory: 2 GB

Problem Statement

As everyone knows, DFS can be used to find cycles in a graph. Some may devise a modified DFS algorithm to find the shortest cycle, as shown in the following pseudocode.

```

visit[node_count] = {false}    // To track visited nodes
depth[node_count] = {0}          // To track depths of nodes
ans = infinity               // Store the length of the shortest cycle

Function DFS(u, p):
    visit[u] = true                // Mark current node as visited
    Sort(adj[u])                 // Lexicographically smallest order

    For each neighbor i of u in adj[u]:
        If i is the parent node (p), skip to the next iteration
        If i is already visited:
            If depth[i] < depth[u]:
                Update ans = min(ans, depth[u] - depth[i] + 1)
        Else:
            Set depth[i] = depth[u] + 1
            Call DFS(i, u)       // Recur for the neighbor


For each vertex i from 1 to n:
    If i isn't visited yet:
        call DFS(i, -1) // Start DFS on vertex i

return ans as the minimum cycle length

```

However, the algorithm is insufficient to find the shortest cycle length in some cases. Your task is to construct a simple graph where the algorithm fails.

Note that when there are multiple choices for the DFS traversal order, the algorithm will choose the lexicographically smallest one. (It will sort all adjacent lists.)

Input

None

Output

Your program should print n, m in the first line, indicating the number of vertices and the number of edges in your graph.

Then print m edges in each line, indicated by two endpoints u_i, v_i of the edge.

Constraints

- $1 \leq n, m \leq 20$

The graph should be a simple graph, meaning it **does not** contain multiple edges or self-loops. Also, the graph should contain **at least one** cycle.

If you outputted an invalid graph, you would also recieve Wrong Answer as result.

Example Output

```
5 6
1 2
1 3
2 4
2 5
3 4
4 5
```

This is an example of incorrect program output, since our algorithm would suffice to find the shortest cycle in this case. The example is provided here just to show the correct output format.