Problem E - Link Protection

In a simple network, there are n servers, and m links. Each link is connecting two servers. Now we want to choose a subset of servers, and put daemons on those servers. If any link failed and it is connected to a server with daemon, the failure can be reported immediately. However, the license fee to the daemons are high. We want as few daemons as possible so that every link is covered. Please write a program to find a valid solution as smaller cost as you can.

Input

First line contains an integer T, denoting the number of test cases.

For each test case, first line contains two integers n, m indicating the number of servers and links respectively.

Then each of the next m lines contains a pair of integers x_i, y_i, denoting a link between server x_i and y_i $(1 \leq x_i, y_i \leq n; x_i \neq y_i)$.

The input is guaranteed that from any server, one can reach any other server by following a sequence of links.

Output

For each test case, first output the size k of the subset. Then output k integers indicating the numbers of servers in a possible solution.

"Aha! That's a simple problem," said Little Tomato, "I doubt that's an NP-complete problem!"

Soon little Tomato came up with a good heuristic — a greedy solution: First choose a server with the largest degree, add it to the set, then remove this server as well as all links connected to this server. Repeat this process until no link remains. If there are several largest-degree servers, choose the one with smallest id.

"I think it is a 2-approximation!" claimed by Little Tomato. That is, using his algorithm it is guaranteed to find a solution with the size of the set of at most twice the size of the minimum solution.

Little Dora thought Little Tomato's solution was wrong, but she cannot find any counterexample. Can you help little Dora to generate such an input file to break Little Tomato's solution?

Input

First line contains an integer T ($1 \leq T \leq 100$), denoting the number of test cases.

For each test case, there is an integer n ($500 \leq n \leq 1000$) indicating the number of servers.

Output

You should output somewhat similar to the input file, that is, begin with T in the first line.

Then for each test case, you need to output its corresponding n, m ($1 \leq m \leq 2500$) first, and output m links: x_i, y_i ($1 \leq x_i, y_i \leq n, x_i \neq y_i$) in the following m lines. There should be at most one link between the same pair of servers, and all servers should be connected.

After these m links, you need to provide a **sample solution**, a valid answer to your test case: begin with k, the size of the subset, then k integers follow denoting the ids of servers. The servers are numbered from 1 to n.

Your output is considered correct, if in all test cases, your sample solution is correct and the solution found by Little Tomato's strategy is more than twice the size of your solution.

Incorrect Sample Input

```
1
3
```

Incorrect Sample Output

```
1
3 3
1 2
1 3
2 3
2 1 2
```

Note

The sample input/output does not satisfy the constraint of problem and even not a correct answer.

Another Note

Little Dora built a monitor to show your effort. She found that this monitor is somewhat magical. It can provide Little Dora any solution and computational power she want! She then names this monitor Dora's Effort Monitor — Dora-e-mon in short.