Problem D. Spanning Tree Removal

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Bob has recently learned algorithms on finding spanning trees and wanted to give you a quiz.

To recap, a spanning tree is a subset of graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected. Given an arbitrary undirected simple graph (without multiple edges and loops), a spanning tree removal is defined as: retrieve one spanning tree of the graph, and remove all the edges selected by the spanning tree from the original graph, obtaining a new graph.

Bob found "spanning tree removal" so fun that he wanted to do it over and over again. In particular, he began with a complete graph, i.e., a graph with every pair of distinct vertices connected by a unique edge. Your goal, is to smartly choose spanning trees to remove so that you can repeat as many times as possible until there is no longer a spanning tree in the graph.

Input

The input file starts with an integer $\mathbf{T}$ ($1 \leq \mathbf{T} \leq 500$), denoting the number of test cases.

Each test case is one line: $\mathbf{N}$ ($2 \leq \mathbf{N} \leq 1000$), which the number of vertices of the graph to begin with.

The sum of $\mathbf{N}$ over all test cases in a single input does not exceed 1000.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number starting from 1, and y is how many times at most you can do the removal.

Then follows $y \times (\mathbf{N} - 1)$ lines. From line $(\mathbf{N} - 1) \times (i - 1) + 1$ to line $(\mathbf{N} - 1) \times i$, you should print a spanning tree you decided to remove at i-th time, in the format that everyone should be familiar with. Namely, each line contains two numbers u and v ($1 \leq u, v \leq \mathbf{N}$, $u \neq v$). (u, v) should be valid tree edge, and does not coincide with edges that have been removed before. If there are several solutions, output any of them.

Example

standard input	standard output
3 2 3 4	Case #1: 1 1 2 Case #2: 1 3 1 1 2 Case #3: 2 1 3 3 4 2 4 3 2 1 4 2 1