code jam

cout << "hello, world!" << endl;

Questions asked 4

Submissions

Old Magician

5pt	Not attempted **203/214 users** correct (95%)
10pt	Not attempted **193/198 users** correct (97%)

Square Fields

10pt	Not attempted **146/157 users** correct (93%)
25pt	Not attempted **107/128 users** correct (84%)

Cycles

15pt	Not attempted **126/146 users** correct (86%)
35pt	Not attempted **20/41 users** correct (49%)

Top Scores

gawry	100
bmerry	100
Olexiy	100
ACRush	100
ardiankp	100
gepa	100
natalia	100
Alexus	100
almelv	100
OpenGL	100

Problem C. **Cycles**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 15 points	Solve C-small
Large input 35 points	Solve C-large

Problem

You are given a complete undirected graph with **n** nodes numbered from 1 to **n**. You are also given **k** *forbidden* edges in this graph.

You are asked to find the number of Hamiltonian cycles in this graph that don't use any of the given **k** edges. A Hamiltonian cycle is a cycle that visits each vertex exactly once. A cycle that contains the same *edges* is only counted once. For example, cycles 1 2 3 4 1 and 1 4 3 2 1 and 2 3 4 1 2 are all the same, but 1 3 2 4 1 is different.

Input

The first line of input gives the number of cases, **N**. **N** test cases follow. The first line of each test case contains two integers, **n** and **k**. The next **k** lines contain two integers each, representing the vertices of a forbidden edge. There will be no self-edges and no repeated edges.

Output

For each test case, output one line containing "Case #**X**: **Y**", where **X** is the case number (starting from 1) and **Y** is the number of Hamiltonian cycles that do not include any of those **k** edges. Print your answer modulo 9901.

Limits

$1 \leq N \leq 10$,
$0 \leq k \leq 15$.

Small dataset

$3 \leq n \leq 10$

Large dataset

$3 \leq n \leq 300$

Sample

```
Input       Output

2           Case #1: 1
4 1         Case #2: 660
1 2
8 4
1 2
2 3
4 5
5 6
```

In the first sample input, there is only one cycle: 1 3 2 4 1.

Powered by

Google Cloud Platform