# Baba learns Topological Sort

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Baba, during his usual training session, came across the following problem on topological sort and was not able to solve it. Hence, he decided to revise the basics of topological sorting once again. Till he is busy revising the basics, can you help him solve this problem ?

Given a DAG (directed acyclic graph) $G$ with $N$ nodes and $M$ edges, among all possible topological sortings of the graph, print the one which is the minimum among all of them if two topological sortings are compared based on the following comparator function :

```
int compare(int A[],int B[],int n){
  //returns 1 if A < B, -1 if A > B, 0 if A == B.
  //A,B : topological sortings.
  for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
      if(A[j] == i && B[j] == i)break;
      else if(A[j] == i)return 1;
      else if(B[j] == i)return -1;

  return 0;
}
```

## Input

First line contains single integer t denoting the number of test cases ($1 \leq t \leq 50$)
For every test case,first line contains two integers $N$ and $M$ ($1 \leq N \leq 10^3, 1 \leq M \leq 2 * 10^3$) denoting the number of nodes and the number of edges in the dag.
Next $M$ lines contain two space separated integers $u$ and $v$ ($1 \leq u, v \leq N$) denoting there is a directed edge from node $u$ to node $v$ in the graph.

## Output

For every test case, print on a new line $N$ space separated integers denoting the minimum topological sorting defined in the question above.

## Example

| standard input | standard output |
|---|---|
| 2 | 2 1 3 4 6 5 |
| 6 5 | 5 1 6 2 3 4 |
| 1 4 | |
| 6 5 | |
| 2 5 | |
| 4 6 | |
| 2 1 | |
| 6 10 | |
| 2 4 | |
| 3 4 | |
| 6 2 | |
| 2 3 | |
| 5 3 | |
| 5 6 | |
| 6 3 | |
| 5 1 | |
| 1 2 | |
| 6 4 | |