I'm lovin' DP!

Description

I'm lovin' DP! Don't you!

DP problems is always the kind of problem which is my top favorite! Once I see a DP problem, I can't help but try to think it, solve it, and enjoy it! When I learn another new trick or skill to solve a DP problem, I can try to apply it to all the DP problem I have seen for the all day! DP problem is so attractive that you can't just skip it!

To make sure you know what's a DP problem, restating it as follow: For those problem which is counting, optimizing, or doing anything about **Disjoint Path**, I call it DP problem!

To let you know how interesting DP problems are, I challenge you the following DP problem:

Given a directed acyclic graph with N vertices, you should choose at most D disjoint paths such that total chosen vertices is maximum.

Directed acyclic graph is a graph where all the edges are directed, and there is no cycle in the graph.

Most importantly, disjoint paths is a set of path such that each vertex belongs to at most one of the path, and each path is valid on the given graph. That is, for a path composed of $v_1, v_2, \ldots, v_k$, there's an directed edge from v_1 to v_2, an directed edge from v_2 to v_3, and so on.

A vertex is chosen if and only if it belongs to one of the path.

Input

The first line contains an integer T indicating the total number of test cases. For each test case, the first line contains three space separated integers N, M, D indicating the number of vertices, the number of edges, and you can choose at most D disjoint paths. Following M lines each contain two space separated integer u_i, v_i indicating that there's a directed edge from u_i to v_i.

- $1 \leq N \leq 514$
- $0 \leq M \leq 1535$
- $1 \leq D \leq N$
- $1 \leq \sum N \leq 6000$
- $1 \leq \sum M \leq 10000$
- $1 \leq u_i \neq v_i \leq N$
- It's guaranteed that the given graph is directed acyclic graph

Output

For each test case, output one integer indicating maximum total chosen vertices.

Sample Input

```
3
2 1 2
1 2
4 3 1
1 2
2 3
2 4
7 6 1
1 4
2 3
3 4
4 5
4 7
5 6
```

Sample Output

```
2
3
5
```