Beautiful Graph Problem

You are given a Graph G with n vertices and m edges. A graph G is beautiful if it does not contain any vertex of degree 1. You can have multiple passes until the graph becomes beautiful. In a single pass, you can delete only vertices of degree 1. Your task is to find the minimum number of passes in order to make the graph beautiful.

Note: Graph G may disconnected.

Input Format

First line contains integer t denoting number of test cases.

In each test case, first line contains integer n and m denoting number of vertices and number of edges. Next m lines contains x and y denoting an edge between vertices x and y.

Constraints

$1 \leq t \leq 16$

$1 \leq n \leq 10^5$

$1 \leq m \leq min(\frac{n(n-1)}{2}, 10^5)$

Output Format

For each testcase, print one line containing one integer representing the minimum number of passes in order to make the graph beautiful.

Sample Input 0

```
1
5 4
1 2
2 3
3 4
4 5
```

Sample Output 0

```
2
```

Explanation 0

Initially, graph is like 1-2-3-4-5 .

In first iteration, delete vertices 1 and 5 . So, graph becomes 2-3-4 .

In second iteration, delete vertices 2 and 4 . So now graph consist of only vertice 3 .

Since, it took us 2 iteration to get rid of all vertices of degree 1, answer is 2 .

Sample Input 1

```
1
3 1
1 2
```

Sample Output 1

```
1
```

Explanation 1

Initially, graph has 2 components, 1---2 and 3. In first iteration , we delete vertices 1 and 2 . So now graph has oly one vertice 3 . So, now ,it has become beautiful . So, answer is 1 .

Submissions: 63
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
```

```
7
8▼      /* Enter your code here. Read input from STDIN. Print output to STDOUT */
9       return 0;
10  }
11
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code