#include <bits/stdc++.h>

using namespace std;

void DFS(int node, vector<bool>& used, vector<vector<int>>& graph, int& count) {

used[node] = 1;

count++;

for (auto& neighbor : graph[node]) {

if (!used[neighbor]) {

DFS(neighbor, used, graph, count);

}

}

}

int main() {

int n, m;

cin >> n >> m;

int res = 0;

int count = 0;

vector<vector<int>> graph(n);

vector<bool> used(n, 0);

for (int i = 0; i < m; i++) {

int u, v;

cin >> u >> v;

graph[u].push\_back(v);

graph[v].push\_back(u);

}

for (int i = 0; i < n; i++) {

if (!used[i]) {

count = 0;

DFS(i, used, graph, count);

if (count >= 5) {

res++;

}

}

}

cout << res << endl;

return 0;

}