#include <cstdio>

#include <cstring>

#include <vector>

using namespace std;

#define inf 1000000000

int n, m;

bool t[2010];

int w[100010], inv[100010];

vector<int> g[2010], e[2010];

void add\_edge(int p, int q, int c)

{

g[p].push\_back(q);

e[p].push\_back(m);

w[m] = c;

inv[m++] = m+1;

g[q].push\_back(p);

e[q].push\_back(m);

w[m] = 0;

inv[m++] = m-1;

}

bool dfs(int p)

{

if (p == (n<<2)+1)

return 1;

int q, ed, i;

for (i = g[p].size()-1; i >= 0; i--)

{

q = g[p][i];

ed = e[p][i];

if (!t[q] && w[ed])

{

t[q] = 1;

if (dfs(q))

{

w[ed]--;

w[inv[ed]]++;

return 1;

}

}

}

return 0;

}

int main()

{

freopen("asteroid.in", "r", stdin);

freopen("asteroid.out", "w", stdout);

int k, p, q, i;

scanf("%d%d", &n, &k);

for (i = 1; i <= n; i++)

{

add\_edge(0, (i<<1)-1, inf);

add\_edge((i<<1)-1, i<<1, 1);

}

for (i = 0; i < k; i++)

{

scanf("%d%d", &p, &q);

add\_edge(p<<1, (n<<1)+(q<<1)-1, inf);

}

for (i = 1; i <= n; i++)

{

add\_edge((n<<1)+(i<<1)-1, (n<<1)+(i<<1), 1);

add\_edge((n<<1)+(i<<1), (n<<2)+1, inf);

}

p = 0;

while (1)

{

memset(t, 0, sizeof(t));

t[0] = 1;

if (!dfs(0))

break;

p++;

}

printf("%d\n", p);

return 0;

}