След поредния скучен работен ден, Лора реши, че е време да основе своя компания. В компанията на Лора има **N** служителя, номерирани от 1 до N, като Лора има номер 1 (разглеждаме и нея като служител).

За да успее една компания, то трябва да има строга йерархия между служителите. Единственото условие за валидна йерархия е всеки служител, освен Лора, да има точно 1 пряк началник, а Лора да няма пряк началник. След дълго психологическо проучване, Лора е направила списък от **M** двойки “Ai Bi”, означаващи, че служител с номер Аi е годен да бъде началник на служител с номер Bi. **За всяка такава двойка е изпълнено Ai < Bi (т.е. номерът на потенциален пряк началник винаги е по-малък от този на подчинения).**

Лора иска да избере йерархията по такъв начин, че броят служители, които не са преки началници на никого е минимален. Помогнете ѝ като напишете програма, която намира минималния брой служители, които не са началници на никого, при оптимална йерархия. Ако няма нито една валидна йерархия – то изведете -1.

**Вход**

От първия ред на файла company.in се въвеждат две цели числа – **N** и **M**, съответно броят служители и броят двойки в списъка на Лора. Следват **M** реда, i-тият от които съдържа две числа разделени с интервал - Ai и Bi – обозначаващи, че служител с номер Ai може да бъде пряк началник на служител с номер Bi.

**Изход**

В изходния файл company.out отпечатайте едно единствено число – минималния брой служители, които не са началници на никого, при оптимално избрана йерархия. Ако няма нито една валидна йерархия отпечатайте -1 на единствен ред.

**Ограничения**

1 ≤ *N* ≤ 30 000

0 ≤ M ≤ 30 000

1 ≤ Ai < Bi ≤ N

**Ограничение за време: 1.0 сек**

**Ограничение за памет: 256 MB**

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (company.in)** | **Изход (company.out)** |
| 3 3  1 2  1 3  2 3 | 1 |
| 6 8  1 2  1 3  1 5  2 5  3 5  2 4  1 4  5 6 | 2 |
| 1 0 | 1 |
| 2 0 | -1 |

**Пояснения**

В първия пример е оптимално 1 да е началник на 2, а 2 на 3. В този случай само 3 не е началник на никого. Алтернативно можехме да сложим 1 да е началник и на 2, и на 3 – но тогава нито 2 нито 3 биха били началници на някого.