

След много труд и усилия (и мазоли по пръстите) Иванчо най-накрая събра пари и си направи собствена софтуерна фирма! Сега обаче се чуди как да окабели офиса, така че всички компютри да имат интернет. За съжаление Иванчо си е купил доста кофти система от Китай, която работи по особен начин. Всеки компютър е свързан с всеки друг с кабел. По всеки кабел може да протича информация само в едната посока. Иванчо иска един от тези компютри да бъде „хъб“, т.е. от него да протича информация към всички останали. Китайците вече са му окабелили офиса, но уви, в него няма хъб. Затова Иванчо се е заел с тежката задача да реконфигурира връзките, така че да получи хъб. За целта той обръща връзките на някои от компютрите (т.е. обръща посоката на протичане на информация по всички кабели свързани с тях – входящите стават изходящи и обратното). Помогнете му, като напишете програма `hub`, която след всяка поредица обръщания на връзки да казва на Иванчо кой от компютрите в мрежата е хъб, ако такъв съществува, или да му казва, че няма такъв.

**Забележка:** Приемаме, че компютрите са номерирани с целите числа от 0 до  $N-1$  включително.

## Вход

От първия ред на файла `hub.in` се въвежда целите числа  $N$  и  $K$  – броят на компютрите в мрежата и броят на заявките за обръщане на върхове.

Следват  $N*(N-1)/2$  реда с по 2 цели числа  $u, v$  обозначаващи еднопосочна връзка от  $u$  към  $v$ .

Следват  $K$  заявки, зададени по следния начин:

Едно цяло число  $P$  – броят на компютрите, чийто връзки ще бъдат обърнати.

$P$  различни цели числа – номерата на компютрите, чийто връзки трябва да бъдат обърнати.

## Изход

В изходния файл `hub.out` отпечатайте по едно цяло число за всяка от  $K$ -те заявки – номерът на компютърът в мрежата, който действа като хъб. Ако такъв не съществува, отпечатайте `-1`.

## Ограничения

$$3 \leq N \leq 850$$

$$0 \leq K \leq 600$$

$$1 \leq P \leq N-1$$

**Ограничение за време: 1.2 сек**

**Ограничение за памет: 256**

## Примерен тест

| Вход (hub.in)                                                                                                         | Изход (hub.out) |
|-----------------------------------------------------------------------------------------------------------------------|-----------------|
| 4 2<br>3 1<br>2 3<br>0 2<br>2 1<br>0 3<br>1 0<br>2<br>2 1<br>2<br>2 0                                                 | -1<br>1         |
| 5 3<br>4 1<br>3 1<br>3 0<br>4 3<br>4 0<br>1 2<br>2 0<br>2 3<br>2 4<br>0 1<br>3<br>2 0 3<br>4<br>1 3 2 0<br>3<br>1 2 4 | -1<br>2<br>3    |

## Пояснения (пример 1)

След първата заявка се обръщат посоките на всички връзки, свързани с компютрите 2 и 1. В получената мрежа няма хъб, за това печатаме -1. След втората заявка, в новополучената мрежа се обръщат всички връзки, свързани с компютрите 2 и 0. Получената мрежа има хъб – компютърът 1.