Даден ни е свързан граф с **N** върха и **M** двупосочни ребра.

Подграф наричаме двойка подмножества от върхове и ребра на даден граф, таковива че краищата на всички ребра в подмножеството от ребра участват в подмножеството от върхове.

Задачата, която ще трябва да решите е следната – съществува ли подграф, отговарящ на следните условия:

1) **Двусвързан** е. Това значи, че при изтриване на който и да е **връх или ребро** от подграфа, той остава свързан.

2) **Двуделен** е. Това значи, че е възможно да оцветим върховете в два цвята, така че краищата на ребрата да са с различен цвят.

3) Броя върхове в подграфа е по-голям или равен на 3.

Напишете програма graph, която проверява дали такъв подграф съществува. Ако това е така, трябва и да намерите пример за такъв подграф.

**Вход**

От първия ред на файла graph.in се въвеждат две числа ***N***и***M***– съответно броя върхове и броя ребра в графа.

От **i**-ят от следващите **M** реда съдържа числата **u[i]** и **v[i]** – информация за съответното ребро от графа. Това значи ,че има ребро между върховете **u[i]** и **v[i]**.

Гарантирано е, че графът е свързан и няма примки **(u[i] ≠ v[i])**. Също така между всяка двойка върхове има най-много едно ребро.

**Изход**

На първия ред на изходния файл graph.out отпечатайте един ред с **„Yes“** или **„No“** - съответно дали съществува подграф с описаните свойства.

Ако такъв подграф съществува, на втория ред отпечатайте две числа **P** и **K** – съответно броя върхове и броя ребра в подграфа. На **i**-я от следващите **K** реда отпечатайтe числата **a[i]** и **b[i]** - крайните върхове на текущото ребро в подграфа. Ако има повече от един такъв подграф, изведете който и да е.

**Ограничения**

**Ограничение за време: 1 сек**

**Ограничение за памет: 256 MB**

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (graph.in)** | **Изход (graph.out)** |
| 4 5  1 2  2 3  3 4  1 3  2 4 | Yes  4 4  1 2 3 4  1 2  1 3  2 4  3 4 |
| 3 3  1 2  2 3  3 1 | No |