**ĐỀ KIỂM TRA LẦN 2 (Very Easy)**

**Tổng quan về các bài thi trong đề**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên bài** | **File**  **dữ liệu** | **File**  **kết quả** | **Điểm** |
| **1** | **Chia đội** | **stdin** | **stdout** | **7,0** |
| **2** | **Thêm cạnh** | **stdin** | **stdout** | **6,0** |
| **3** | **Xếp xe** | **stdin** | **stdout** | **7,0** |

*Phần mở rộng của File chương trình là PAS hoặc CPP tùy theo ngôn ngữ lập trình sử dụng là Pascal hoặc C++*

***Viết chương trình giải các bài toán sau:***

# Bài 1. Chia đội

Ở một giải đấu có n trận đấu được sắp xếp diễn ra theo thứ tự lần lượt từ 1 đến m, trận đấu thứ i là của 2 võ sĩ u­­i và vi (ui ≠ vi).

Tuy nhiên sau khi sắp xếp xong lịch thi đấu, bỗng vì vài lý do mà thể thức thay đổi. Khi này, các võ sĩ sẽ chia làm 2 đội có số người bằng nhau và thực hiện lần lượt các cặp đấu như đã sắp xếp, nhưng lại có 1 quy định mới là 2 người cùng 1 đội sẽ không được đấu với nhau! Điều này đồng nghĩa với việc sẽ có một vài trận đấu bị loại bỏ. Nhưng để tránh phải sắp xếp lại lịch, BTC quyết định sẽ giữ lại i trận đấu đầu tiên và loại bỏ hết các trận đấu sau đó sao cho đảm bảo rằng tồn tại cách chia đội để không một võ sĩ nào đấu nhau.

Lưu ý rằng khi đó, sẽ có thể có một vài võ sĩ không được đấu một trận nào, điều này không vấn đề gì.

Ban tổ chức muốn tổ chức nhiều trận đấu nhất để tối ưu hóa lợi nhuận, bạn hãy tính xem số trận tối đa có thể tổ chức là bao nhiêu.

**Input:**

* Dòng đầu tiên là 2 số nguyên dương **n,m** () lần lượt là số võ sĩ và số trận đấu. Đầu vào luôn đảm bào **n** chẵn.
* **m** dòng tiếp theo, mỗi dòng chứa 2 số nguyên dương u,v ().

**Output:**

* In ra 1 số nguyên dương là số trận tối đa có thể đấu.

**Subtask:**

* **Subtask 1 (20% điểm): n, m ≤ 5.**
* **Subtask 2 (30% điểm): n, m ≤ 1000.**
* **Subtask 3 (50% điểm): không giới hạn nào thêm.**

|  |  |
| --- | --- |
| **stdin** | **stdout** |
| 6 6  1 4  1 5  2 4  1 3  2 6  3 6 | 4 |

Có thể chia 2 đội: (1,2,6) và (3,4,5). Nếu chọn thêm trận thứ 5 thì chỉ có 1 cách: (1,2) và (3,4,5,6) => không thỏa mãn do 2 đội không cân bằng số người.

# Bài 2. Thêm cạnh

Cho một đồ thị dạng cây có n đỉnh, độ dài đường đi giữa 2 đỉnh bất kỳ được tính bằng số cạnh trên đường đi đơn giữa 2 đỉnh đó.

Có q truy vấn, mỗi truy vấn sẽ cho bạn 2 cặp đỉnh u, v, t1, t2. Câu trả lời của truy vấn này là hãy tính độ dài đường đi ngắn nhất từ u đến v nếu có thêm cạnh nối t1, t2.

**Input:**

* Dòng đầu tiên là 2 số nguyên dương n, q
* n- 1 dòng tiếp theo chứa các cặp số biểu diễn cho cạnh của cây.
* q dòng sau đó, mỗi dòng chứa 4 số nguyên u, v, t1, t2 là thông tin mỗi truy vấn.

**Output**:

Với mỗi truy vấn, in kết quả ra 1 dòng.

|  |  |
| --- | --- |
| **stdin** | **stdout** |
| 5 2  1 2  1 3  2 4  2 5  4 3 2 3  4 5 4 5 | 2  1 |

**Subtask:**

* Subtask 1 (40% số điểm) : .
* Subtask 2 (60% số điểm) : Không có giới hạn nào thêm.

# Bài 3. Xếp xe

Trung tâm giao vận cần lập lộ trình cho nhân viên giao hàng để phục vụ yêu cầu vận chuyển của n khách hàng. Khách hành i có điểm đón hàng là i và điểm trả hàng là i+n. Như vậy, một lộ trình là hoán vị của (1,2,…n, n+1, n+2,…, 2n) trong đó điểm lấy hàng i luôn phải ở trước điểm nhận hàng i+n.

Hỏi có tất cả bao nhiêu lộ trình có thể thỏa mãn ? Do con số rất lớn nên bạn chỉ cần in ra phần dư của kết quả sau khi chia cho 109 + 7.

**Input:**

* Dòng đầu tiên chứa 1 số nguyên dương n. ().

**Output:**

* In ra một số nguyên duy nhất là đáp án của bài.

**Subtask:**

* 20% số điểm của bài ứng với .
* 30% số điểm của bài ứng với 6 ≤ n ≤ 10.
* 50% còn lại không ràng buộc gì thêm.

|  |  |
| --- | --- |
| **stdin** | **stdout** |
| 2 | 6 |
| 1 | 1 |
| 5 | 113400 |

**---HẾT---**