**Productive productivity**

**( <https://www.hackerearth.com/practice/algorithms/graphs/minimum-cost-maximum-flow/practice-problems/algorithm/productive-productivity/> )**

Little Jhool là một trợ lý giảng dạy rất khoan dung trong trường đại học của mình. Anh ấy không thích cắt điểm của sinh viên, vì vậy, rõ ràng, mọi sinh viên trong hướng dẫn của anh ấy đều yêu anh ấy. Đặc biệt, các cô gái. Họ phát điên vì anh ta.

Nhưng dù sao, giáo viên cũng phải biết về sự khoan hồng của Jhool trong khi chấm điểm, vì vậy lần này trong kỳ thi, anh quyết định đưa ra một bài kiểm tra khác nhau cho mỗi học sinh để kiểm tra xem học sinh được Jhool dạy tốt như thế nào. Bây giờ, Little Jhool biết các chủ đề mạnh và yếu của mỗi học sinh, vì vậy anh muốn tối đa hóa tổng số điểm mà học sinh đạt được trong hướng dẫn của mình.

Bạn được cung cấp số lượng sinh viên trong hướng dẫn của Jhool, ký hiệu là **n** - **n** cũng là số lượng bài thi khác nhau - nghĩa là, mỗi bài cho mỗi sinh viên. Mỗi học sinh sẽ chỉ nhận được một bài thi để giải. Bạn còn được cung cấp một ma trận, (nxn) biểu thị các điểm mà mọi học sinh sẽ nhận được nếu anh ta thử một bài kiểm tra cụ thể. Bạn đã giúp Jhool tìm ra một cách mà anh ta có thể tối đa hóa tổng số điểm mà cả lớp đạt được.

**Input:**  
Dòng đầu tiên chứa số trường hợp thử nghiệm **T** . Dòng thứ hai chứa một số nguyên, biểu thị n , theo sau là ma trận n X n - biểu thị điểm số tương ứng của mỗi học sinh trong kỳ thi tương ứng đó.

**Output:**   
Tìm điểm **TOTAL** tối đa thu được của tất cả các sinh viên cộng lại.

**Các ràng buộc:**  
1 <= **T** <= 20   
1 <= **N** <= 100   
1 <= A i j <= 1000

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  3  1 2 3  4 5 6  7 8 9 | 15 |