

Đội tuyển dự thi Olympiad có n học sinh. Mỗi học sinh thứ i được đặc trưng bởi 2 tham số: hệ số kỹ năng a_i và chỉ số thông minh b_i .

Giờ học được tiến hành theo kiểu sau: Giáo viên phụ trách đội tuyển lần lượt làm việc với học sinh, thảo luận, giải quyết những vấn đề nảy sinh. Kết quả là hệ số kỹ năng của học sinh được tăng thêm một lượng bằng chỉ số thông minh. Như vậy học sinh càng thông minh bao nhiêu thì hệ số kỹ năng càng tăng nhiều bấy nhiêu.

Do hạn chế về thời gian, trong suốt quá trình học, giáo viên chỉ có thể thực hiện được c lần làm việc riêng với học sinh. Một học sinh có thể được làm việc nhiều lần với giáo viên.

Số liệu thống kê cho thấy, muốn được giải trong kỳ thi thì hệ số kỹ năng phải không nhỏ hơn k .

Yêu cầu: Cho các số nguyên n, c, k và a_i, b_i ($1 \leq n \leq 10^6, 1 \leq c, k \leq 10^9, 0 \leq a_i, b_i \leq 10^9, i = 1 \div n$). Hãy xác định số lượng tối đa học sinh sẽ được giải.

Dữ liệu: Vào từ file văn bản OLYMP.INP:

- Dòng đầu tiên chứa 3 số nguyên n, c và k ,
- Dòng thứ i trong n dòng sau chứa 2 số nguyên a_i và b_i .

Kết quả: Đưa ra file văn bản OLYMP.OUT một số nguyên – số lượng tối đa học sinh sẽ được giải.

Ví dụ:

OLYMP.INP	OLYMP.OUT
3 5 6	2
1 1	
2 1	
4 2	

TRẢ NỢ

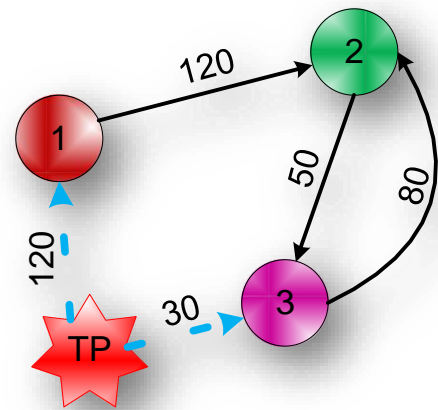
DEBT.???

Ở một thị trấn nhỏ có n người sống, đánh số từ 1 đến n . Mỗi người vay *đúng một người khác* một khoản tiền. Năm hết tết đến, mọi người tìm cách thanh toán hết nợ nần để năm mới gặp may mắn. Đáng tiếc là không ai có đủ tiền mặt để trả nợ.

Chính quyền thành phố quyết định hỗ trợ cho một số người vay một ít tiền. Những người này sẽ trả được nợ của mình và tạo phản ứng dây chuyền: **A** trả cho **B**, **B** trả cho **C**, . . . Nếu người **B** chưa đủ trả cho **C** thì phải chờ đợi bao giờ có đủ mới trả. Nếu sau khi trả nợ vẫn còn thừa tiền thì **B** sẽ giữ lại số tiền thừa.

Chính quyền không muốn chi quá nhiều tiền. Thị trưởng yêu cầu tính toán số tiền nhỏ nhất cần hỗ trợ để mọi người thanh toán hết nợ của mình.

Ví dụ, với $n = 3$, người 1 nợ người 2 số tiền là 120, người 2 nợ người 3 số tiền là 50, số tiền người 3 phải trả cho người 2 là 80. Khi đó chính quyền chỉ cần cho người một vay 120, người 3 – 30, mọi món nợ trong dân chúng sẽ được giải quyết. Như vậy chính quyền phải chi tổng cộng là 150.



Yêu cầu: Cho n , a_i , b_i ($2 \leq n \leq 200\,000$, $1 \leq b_i \leq 10\,000$, $1 \leq a_i \leq n$, $i = 1 \div n$), trong đó a_i là chủ nợ của người i và số tiền người i phải trả là b_i ($a_i \neq i$).

Hãy xác định số tiền nhỏ nhất chính quyền cần hỗ trợ.

Dữ liệu: Vào từ file văn bản DEBT.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa 2 số nguyên a_i và b_i .

Kết quả: Đưa ra file văn bản DEBT.OUT một số nguyên – số tiền nhỏ nhất chính quyền cần hỗ trợ.

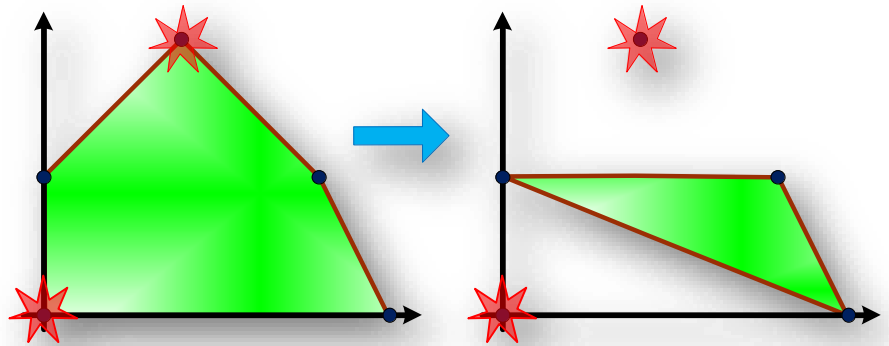
Ví dụ:

DEBT.INP	DEBT.OUT
3	150
2 120	
3 50	
2 80	

ĂN CỬ QUÂN SỰ TRONG RỪNG RẬM

JUNGLE.???

Căn cứ quân sự được thiết lập sâu trong rừng rậm. Có n trạm gác đặt máy phát siêu âm để bảo vệ căn cứ, trạm thứ i có tọa độ (x_i, y_i) . Không có 3 trạm nào thẳng hàng. Nếu lấy các trạm này làm đỉnh ta có một đa giác lồi. Không có trạm nào nằm trong đa giác lồi. Vùng nằm hoàn toàn trong đa giác lồi được bảo vệ an toàn tuyệt đối. Nhưng kẻ địch có thể tấn công và phá hủy một số trạm gác. Với những trạm còn lại, vùng được bảo vệ sẽ bị thu hẹp.



Bộ chỉ huy căn cứ cần phải được đảm bảo an toàn ở mức cao nhất, sao cho để Bộ chỉ huy không còn được an toàn thì phải phá hủy một số nhiều nhất các trạm gác.

Yêu cầu: Cho n và các tọa độ nguyên x_i, y_i ($3 \leq n \leq 5 \times 10^4$, $|x_i|, |y_i| \leq 10^6$, $i = 1 \div n$). Hãy xác định số trạm gác tối đa cần phá để Bộ chỉ huy không còn được an toàn ứng với trường hợp vị trí đặt bộ chỉ huy được chọn tối ưu.

Dữ liệu: Vào từ file văn bản JUNGLE.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa 2 số nguyên x_i, y_i , các đỉnh được liệt kê theo chiều kim đồng hồ.

Kết quả: Đưa ra file văn bản JUNGLE.OUT một số nguyên – số trạm gác tối đa cần phá để Bộ chỉ huy không còn được an toàn.

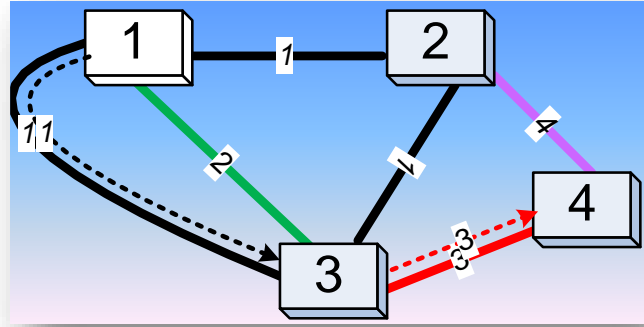
Ví dụ:

JUNGLE.INP	JUNGLE.OUT
5	2
0 0	
0 10	
10 20	
20 10	
25 0	

ĐƯỜNG ĐI LÝ TƯỞNG

Tên chương trình: IDEAL.???

Mê cung có n phòng đánh số từ 1 đến n , được nối với nhau bởi m đoạn đường đi 2 chiều nối trực tiếp 2 phòng, đường đi thứ i có màu c_i . Giữa 2 phòng có thể có nhiều đường đi nối trực tiếp. Đường đi trực tiếp có thể nối một phòng với chính nó. Người chơi được máy bay lên thẳng thả xuống phòng 1 và phải tìm đường đi tới phòng n . Độ dài của đường đi được tính bằng số đoạn đường nối giữa 2 phòng đã đi qua. Ai có đường đi ngắn nhất sẽ thắng cuộc. Nếu có nhiều người cùng có đường đi ngắn nhất thì so sánh theo thứ tự từ điển các màu đã đi qua. Người có đường đi ngắn nhất với thứ tự từ điển nhỏ nhất sẽ thắng. Đường đi lý tưởng là đường ngắn nhất và có thứ tự từ điển nhỏ nhất.



Yêu cầu: Cho m, n và các đường đi (a_i, b_i, c_i) xác định đường màu c_i nối từ phòng a_i tới phòng b_i ($2 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5, 1 \leq a_i, b_i \leq n, 1 \leq c_i \leq 10^9$). Dữ liệu đảm bảo có đường đi từ phòng 1 đến phòng n . Hãy xác định đường đi lý tưởng: số đoạn đường và màu của các đoạn đó theo trình tự đi.

Dữ liệu: Vào từ file văn bản IDEAL.INP:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Dòng thứ i trong m dòng sau chứa 3 số nguyên a_i, b_i và c_i .

Kết quả: Đưa ra file văn bản IDEAL.OUT:

- Dòng đầu tiên chứa số nguyên k ,
- Dòng thứ 2 chứa k số nguyên – màu của các đoạn theo trình tự đi.

Ví dụ:

IDEAL.INP		
4	6	
1	2	1
1	3	2
3	4	3
2	3	1
2	4	4
3	1	1

IDEAL.OUT	
2	
1	3