

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		
3	5	6
1	1	
2	1	
4	2	

OLYMP.OUT	
2	

TRẢ NỢ

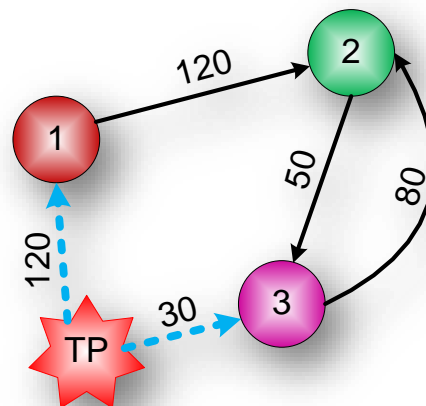
Tên chương trình: DEBT.???

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 trường 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 1 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	
3	
2	120
3	50
2	80

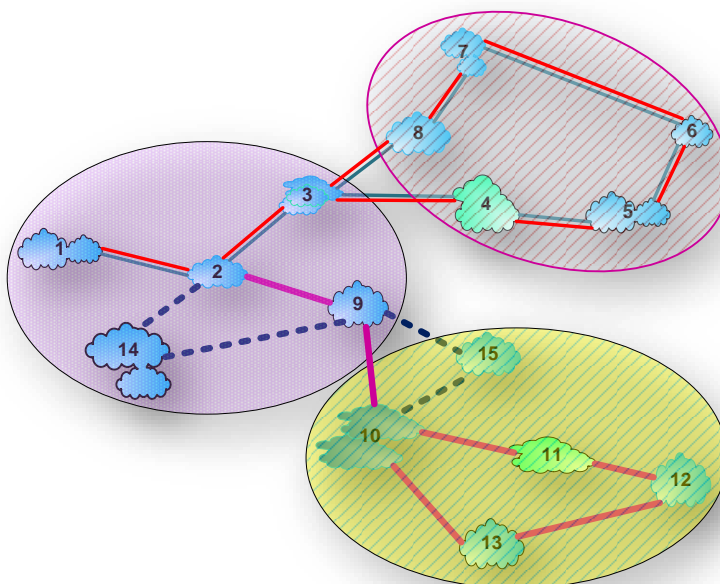
DEBT.OUT
150

CÁCH MẠNG XƯƠNG RỒNG

Tên chương trình: CACTUS.???

Thành phố ngầm dưới đất hình thành sau chiến tranh hạt nhân tổng lực toàn cầu. Thành phố có n hang ngầm, nơi người dân sinh sống, đánh số từ 1 đến n . Có m tuyến đường ngầm nối các hang. Mỗi tuyến đường có thể đi qua một hang nhiều lần, nhưng giữa 2 hang có không quá một đường nối trực tiếp.

Một cuộc cách mạng nổ ra lật đổ chính quyền hiện hành, nhưng k chính đảng tồn tại không thỏa thuận được với nhau để thành lập chính phủ liên hiệp và cuối cùng họ đi đến thỏa hiệp chia toàn bộ thành phố thành k quận, mỗi quận có số hang như nhau, các hang trong một quận đi đến được với nhau không phải qua hang nào ở quận khác, mỗi quận sẽ sống với luật lệ riêng của mình. May mắn là n chia hết cho k .



Yêu cầu: Cho n , m , k và dãy các hang theo trình tự nối ở mỗi tuyến ($1 \leq n \leq 5 \times 10^4$, $0 \leq m \leq 10^4$, $1 \leq k \leq n$). Ở

các tuyến không có đoạn đường nối nào được nêu quá một lần. Hãy xác định hang cho mỗi quận và nêu thành danh sách n/k số theo thứ tự tăng dần. Nếu không có cách chia thỏa mãn thì đưa ra số -1.

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

- Dòng đầu tiên chứa 3 số nguyên n , m và k ,
- Mỗi dòng trong m dòng sau nêu danh sách các hang theo trình tự nối của mỗi tuyến, bắt đầu bằng một số nguyên – số lượng hang trong tuyến.

Kết quả: Đưa ra file văn bản CACTUS.OUT số -1 hoặc k dòng, mỗi dòng có n/k số nguyên (theo trình tự tăng dần) xác định một quận.

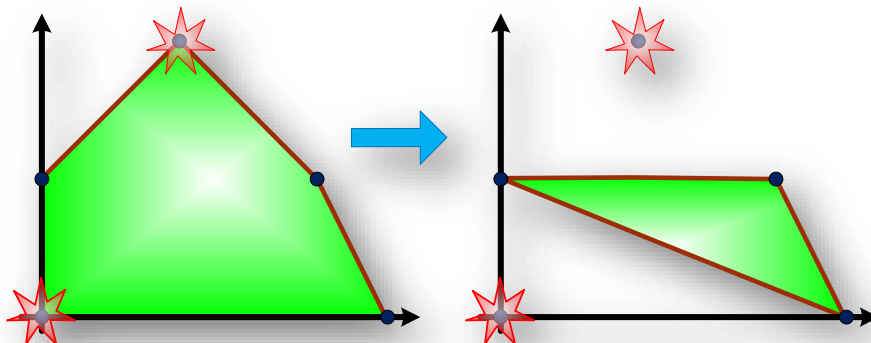
Ví dụ:

CACTUS.INP	CACTUS.OUT
15 3 3	4 5 6 7 8
9 1 2 3 4 5 6 7 8 3	10 11 12 13 15
7 2 9 10 11 12 13 10	1 2 3 9 14
5 2 14 9 15 10	

CĂN CỨ QUÂN SỰ TRONG RỪNG RẬM

Tên chương trình: 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 địch 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
5
0 0
0 10
10 20
20 10
25 0

JUNGLE.OUT
2

