

1. Tường bảo vệ

Tại một mục tiêu quan trọng người ta lắp một màn hình điện tử dưới dạng hình chữ nhật kích thước $M \times N$ ô vuông. Mỗi ô của màn hình có thể ở một trong hai trạng thái: tắt hoặc sáng. Khi dùng bút laze kích vào một ô nào đó của màn hình thì một số ô sẽ chuyển sang trạng thái (từ sáng sang tắt và ngược lại) theo mẫu kích thước 3×3 cho trước. Các ô còn lại không thay đổi trạng thái. Ở mẫu bên, các ô tô đậm đổi trạng thái. Ô trên màn hình được kích hoạt luôn là tâm của mẫu.

Ban đầu tất cả các ô của màn hình đều ở trạng thái tắt. Bằng cách kích hoạt một số ô, có thể người ta sẽ đưa tất cả các ô của màn hình sang trạng thái sáng.

Ví dụ, nếu màn hình có kích thước 2×3 thì trình tự kích hoạt

là 2, 5, 6

1	2	3
4	5	6

1	2	3
4	5	6

1	2	3
4	5	6

Yêu cầu: Ghi ra trình tự kích hoạt ít ô nhất để chuyển tất cả các ô màn hình sang trạng thái sáng. Các ô của màn hình được đánh số theo hàng, từ 1 đến $M \times N$.

Dữ liệu: Vào từ file văn bản PANEL.INP chứa dữ liệu theo qui cách sau:

- Dòng đầu tiên chứa hai số $M \times N$ ($1 \leq M, N \leq 100$)
- Ba dòng sau, mỗi dòng là một xâu 3 ký tự: ký tự '*' chỉ ô đổi trạng thái, ký tự '.' thể hiện ô không đổi trạng thái

Kết quả: Ghi ra file văn bản PANEL.OUT.

- Dòng đầu tiên chứa số lượng ô cần kích hoạt (-1 nếu không có cách kích hoạt)
- Nếu có cách kích hoạt thì dòng tiếp theo ghi số hiệu các ô sẽ lần lượt kích hoạt để đưa bảng về trạng thái sáng

Ví dụ:

PANEL.INP	PANEL.OUT
2 3 * * . . * . * . .	3 2 5 6

2. XÂY DỰNG SÂN BAY

Nhằm phát triển du lịch, chính phủ quyết định sẽ xây dựng một sân bay mới trên cao nguyên Đà Lạt. Đây là khu vực núi non nên cần phải tính toán thật hợp lý. Để đảm bảo những máy bay lớn như Airbus A380 có thể hạ cánh an toàn, đường băng phải đảm bảo nằm ngang và độ dài ít nhất là L . Người ta có thể phá những quả núi lớn để tạo những khu đất phẳng, nhưng việc đắp thêm đất là vô cùng nguy hiểm vì kết cấu đặc biệt của đất ở nơi này. Bạn cần tính xem cần phải phá đi lượng đá tối thiểu là bao nhiêu để có thể xây được sân bay.

Dữ liệu vào là file AIRPORT.INP:

- Dòng thứ nhất ghi 2 số nguyên N và L .
- N dòng tiếp theo, mỗi dòng ghi tọa độ x, y thể hiện địa hình nơi đây. Các số xi được đảm bảo sắp xếp tăng ngặt.

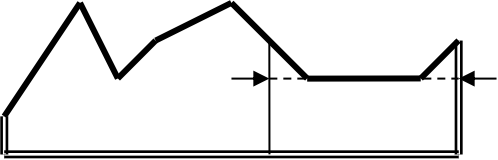
Dữ liệu ra file AIRPORT.OUT:

- Ghi ra duy nhất lượng đất cần phải phá để xây được sân bay với sai số là 0.001.

Giới hạn:

- $1 \leq N \leq 500$.
- $1 \leq L \leq 10000$.
- $0 \leq x_i, y_i \leq 10000$.

Ví dụ:

AIRPORT.INP	AIRPORT.OUT	Giải thích
8 5 0 1 2 5 3 3 4 4 6 5 8 3 11 3 12 4	1.000	

3. Đảo tham lam

Trên đường đi tìm cha, cậu bé Gon lạc đến đảo Tham Lam. Hòn đảo này rất kỳ quái, người dân không dùng tiền mà dùng thẻ để trao đổi. Trên mỗi thẻ ghi một số nguyên nằm trong khoảng $[1, N]$ và được gọi là mã số của thẻ. Chỉ có một cách duy nhất để ra khỏi đảo là đem được N thẻ có mã đôi một khác nhau (từ là mã $1, 2, \dots, N$) đổi lấy vé tàu.

Gon có 2 cách để kiếm thẻ ở trên đảo:

- Nhặt thẻ mà người khác đánh rơi
- Trao đổi với ngân hàng của đảo: dùng 1 thẻ của mình đổi lấy 1 thẻ khác của ngân hàng, lệ phí 1 lần đổi là 1 cục vàng (ngân hàng dùng để đúc thẻ mới)

Rất may là Gon được một người bạn tốt bụng tặng cho N thẻ nên chỉ còn phải ra ngân hàng đổi thẻ (để được N thẻ khác nhau đôi một). Vì chuyến đi dài nên Gon phải tiết kiệm vàng. Bạn hãy giúp Gon tìm cách đổi để tốn ít vàng nhất mà vẫn đổi được vé tàu đi ra khỏi đảo. Biết rằng luôn tồn tại ít nhất một cách đổi.

Dữ liệu: Vào từ file GREED.IN

- Dòng thứ nhất ghi số nguyên N ($N \leq 100$)
- Dòng thứ hai ghi N số nguyên là mã số của N thẻ mà Gon có
- Tiếp theo là một số dòng, trên mỗi dòng chứa hai số u, v có nghĩa là có thể đổi thẻ có mã số u của Gon lấy thẻ có mã số v của ngân hàng và ngược lại.

Các dữ liệu số trên cùng một dòng được ghi cách nhau bởi ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản GREED.OUT một số nguyên duy nhất là tổng số vàng ít nhất phải trả khi đổi thẻ

Ví dụ:

GREED.IN	GREED.OUT
4	4
1 1 1 1	
1 2	
2 3	
1 4	
3 4	