

TẬP KẾT

Một bàn cờ kích thước $m \times n$ trong đó đánh dấu một số ô cấm. Các hàng của bàn cờ được đánh số từ 1 tới m theo thứ tự từ trên xuống dưới và các cột của bàn cờ được đánh số từ 1 tới n theo thứ tự từ trái qua phải. Trên bàn cờ có k quân mã đang đứng ở những vị trí khác nhau. Cần đi những quân mã này đến k vị trí tập kết khác nhau, mỗi quân mã một vị trí. Trong quá trình di chuyển, mỗi bước mã phải nhảy theo một nước đi của quân mã theo luật cờ, không được nhảy đến các ô cấm và không được nhảy đến ô đang có quân mã khác đang đứng. Vai trò của các quân mã và các vị trí tập kết là như nhau (một quân mã có thể cho đi tới bất kỳ vị trí tập kết nào nếu có đường nhảy).

Yêu cầu: Lập chương trình xác định cách đi các quân mã sao cho tổng số bước đi của các quân mã là nhỏ nhất.

Dữ liệu: Vào từ file văn bản KNIGHTS.INP

- Dòng 1: Ghi hai số nguyên dương m, n
- m dòng tiếp theo, dòng i ghi n ký tự thể hiện hàng i của bàn cờ. Ký tự thứ i có thể là:
 - “.”: Thể hiện ô trống
 - “#”: Thể hiện ô cấm
 - “\$”: Thể hiện ô có mã đang đứng
 - “@”: Thể hiện ô ở vị trí tập kết

Kết quả: Ghi ra file văn bản KNIGHTS.OUT

- Dòng 1 ghi tổng số bước di chuyển để đưa các quân mã đến vị trí tập kết (p)
- p dòng tiếp theo liệt kê các bước di chuyển theo thứ tự thực hiện: mỗi dòng ghi bốn số nguyên x_1, y_1, x_2, y_2 cách nhau đúng một dấu cách, cho biết bước di chuyển tương ứng sẽ chuyển quân mã ở ô (x_1, y_1) tới ô (x_2, y_2)

Ràng buộc:

$m, n, k \leq 100$; Dữ liệu vào luôn đảm bảo có phương án thực hiện yêu cầu đề ra. Các số trên một dòng của input/output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ:

KNIGHTS.INP	KNIGHTS.OUT
6 6	7
#.#..\$	5 4 4 6
...\$@.	4 6 2 5
..#...	4 3 5 5
..\$#..	1 6 3 5
##.\$@@	3 5 5 6
#....@	2 4 4 5
	4 5 6 6

DÃY SỐ

Cho một dãy gồm n số nguyên $A = (a_1, a_2, \dots, a_n)$ và một số nguyên k . Hãy xác định xem trong dãy A có tồn tại hai phần tử a_p, a_q ở hai vị trí khác nhau $p \neq q$ mà $a_p - a_q = k$ hay không.

Dữ liệu: Vào từ file văn bản SEQ.INP

- Dòng 1: Chứa hai số nguyên n và k ($2 \leq n \leq 10^5, |k| \leq 2 \cdot 10^9$)
- Dòng 2: Chứa n số nguyên a_1, a_2, \dots, a_n ($\forall i: |a_i| \leq 2 \cdot 10^9$)

Kết quả: Ghi ra file văn bản SEQ.OUT hai chỉ số p, q tìm được. Nếu không tồn tại cặp số thỏa mãn yêu cầu, ghi ra hai số 0

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ:

SEQ.INP	SEQ.OUT
7 88	7 1
11 33 55 99 33 77 99	

XỔ SỐ

Công ty điện thoại LuckyPhone tổ chức xổ số để quyên góp tiền cho quỹ hỗ trợ tài năng trẻ nước NumberLand. Luật chơi như sau:

- Có n khách hàng tham gia chơi ($1 \leq n \leq 10^6$) mỗi khách hàng tự chọn một số tự nhiên có không quá 9 chữ số, sau đó soạn tin nhắn gửi số mình chọn đến số điện thoại 19001234. Mỗi khách hàng chỉ được gửi đúng một tin nhắn.
- Trong các số nhận được từ các khách hàng, số lớn nhất trong các số được ít khách hàng gửi đến nhất sẽ được chọn làm số may mắn (Lucky Number).
- Công ty thu từ mỗi khách hàng đã gửi tin nhắn x đồng ($1 \leq x \leq 10^9$) và phải tặng cho tất cả các khách hàng đã gửi số may mắn mỗi người một giải thưởng giá trị đúng bằng số may mắn.
- Lợi nhuận của công ty thu được trong cuộc chơi bằng tổng số tiền thu được từ các khách hàng trừ đi số tiền trao giải thưởng.

Yêu cầu: Hãy tính lợi nhuận của công ty LuckyPhone trong cuộc chơi

Dữ liệu: Vào từ file văn bản LOTTERY.INP

- Dòng 1: Chứa hai số nguyên dương n, x
- n dòng tiếp theo, mỗi dòng chứa một số nhận được từ một khách hàng

Kết quả: Ghi ra file văn bản LOTTERY.OUT lợi nhuận thu được

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ:

LOTTERY.INP	LOTTERY.OUT
9 100	12
2	
33	
1	
2	
444	
33	
1	
1	
444	

Giải thích: Trong ví dụ trên, có 3 số ít được chọn nhất là 2, 33 và 444 (mỗi số được chọn 2 lần). Theo quy tắc đặt ra, số may mắn là số 444. Tổng tiền thu được từ khách hàng là $9 \times 100 = 900$, tổng tiền trả thưởng là $2 \times 444 = 888$. Vậy lợi nhuận của công ty trong cuộc chơi là $900 - 888 = 12$

LẮP RÁP Ô TÔ

Để tự động hoá dây chuyền lắp ráp ô tô, người ta sử dụng một robot lắp ráp và n dụng cụ lắp ráp đánh số từ 1 tới n . Có tất cả m loại bộ phận trong một chiếc ô tô đánh số từ 1 tới m . Mỗi chiếc ô tô phải được lắp ráp từ t bộ phận $O = (o_1, o_2, \dots, o_t)$ theo đúng thứ tự này ($1 \leq o_i \leq m, \forall i$). Biết được những thông tin sau:

- Tại mỗi thời điểm, robot chỉ có thể cầm được 1 dụng cụ.
- Tại thời điểm bắt đầu, robot không cầm dụng cụ gì cả và phải chọn một trong số n dụng cụ đã cho, thời gian chọn không đáng kể.
- Khi đã có dụng cụ, robot sẽ sử dụng nó để lắp một bộ phận trong dãy O , biết thời gian để Robot lắp bộ phận loại v bằng dụng cụ thứ i là b_{iv} ($1 \leq i \leq n; 1 \leq v \leq m$)
- Sau khi lắp xong mỗi bộ phận, robot được phép đổi dụng cụ khác để lắp bộ phận tiếp theo, biết thời gian đổi từ dụng cụ i sang dụng cụ j là a_{ij} . (Lưu ý rằng a_{ij} có thể khác a_{ji} và a_{ii} luôn bằng 0).

Yêu cầu: Hãy lập trình cho robot lắp ráp chiếc ô tô từ các bộ phận $O = (o_1, o_2, \dots, o_t)$ trong thời gian ngắn nhất.

Dữ liệu: Vào từ file văn bản VITERBI.INP

- Dòng 1: Chứa 3 số nguyên dương $n, m, t \leq 200$
- Dòng 2: Chứa t số nguyên dương o_1, o_2, \dots, o_t
- n dòng tiếp theo, dòng thứ i chứa n số nguyên $a_{i1}, a_{i2}, \dots, a_{in} \leq 200$
- n dòng tiếp theo, dòng thứ j chứa m số nguyên $b_{j1}, b_{j2}, \dots, b_{jm} \leq 200$

Kết quả: Ghi ra file văn bản VITERBI.OUT một số nguyên duy nhất là thời gian ít nhất để lắp ráp xong toàn bộ t linh kiện $O = (o_1, o_2, \dots, o_T)$.

Các số trên một dòng của Input files được ghi cách nhau ít nhất một dấu cách

Ví dụ:

VITERBI.INP	VITERBI.OUT
3 4 8	21
1 2 3 4 1 2 3 4	3 2 1 1 3 2 1 1
0 9 1	
1 0 9	
9 1 0	
8 8 1 5	
8 1 8 8	
1 8 8 5	