

Tặng hoa kiểu úc

Nhân ngày 8/3, một bạn nam trong lớp muốn tặng hoa cho một bạn nữ mà sở thích của bạn nữ này kỳ quặc đến mức chỉ có... máy tính mới hiểu được. Chẳng hạn bạn nữ này cho rằng trong bó hoa được tặng, đã có hoa hồng thì phải có hoa cúc, đã có hoa cúc thì phải có hoa phăng, mà đã có hoa phăng thì lại phải có... hoa hồng. Và nếu như ai đem tặng cô ta một bó hoa không ưng ý thì thà không tặng còn hơn bởi hậu quả ra sao thì cũng chỉ có... máy tính mới hiểu được.

Yêu cầu: Hãy chọn một bó hoa gồm ít loại hoa nhất mà vẫn phù hợp với sở thích của bạn nữ khó tính đó

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

- Dòng 1 ghi N là số lượng các loại hoa ($1 \leq N \leq 200$)
- Các dòng tiếp theo, mỗi dòng ghi hai số u và v cho biết nếu đã tặng loại hoa u thì sẽ phải tặng luôn loại hoa v

Kết quả: Ghi ra file văn bản OFFER.OUT ghi số nguyên dương k là số loại hoa chọn ra được

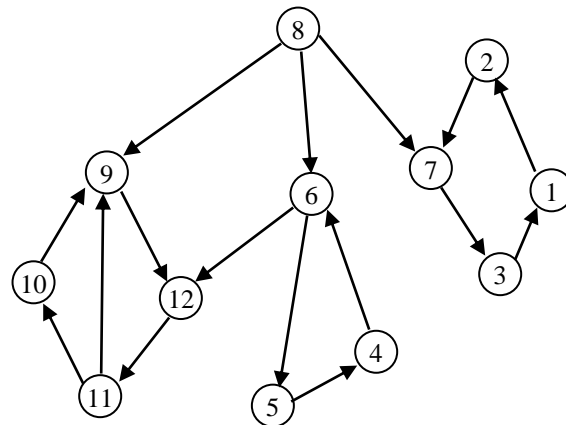
Ví dụ:

OFFER.INP

```
12
1 2
2 7
3 1
4 6
5 4
6 5
6 12
7 3
8 6
8 7
8 9
9 12
10 9
11 9
11 10
12 11
```

OFFER.OUT

```
4
GT: Chọn
9, 10, 11, 12
```



Bộ sưu tập có giá trị (collect.cpp)

Một bộ sưu tập tiền xu cổ được coi là có giá trị phải gồm không ít hơn Z_0 đồng tiền vàng, S_0 đồng tiền bạc và M_0 đồng tiền đồng. Bộ sưu tập ban đầu của Alibaba có một số lượng nhất định các đồng tiền vàng bạc và đồng nhưng chưa phải là một bộ sưu tập có giá trị. Tại Trụ sở của hiệp hội những người sưu tập tiền cổ có đặt một máy đổi tiền để giúp hội viên đổi được các bộ sưu tập có giá trị. Tuy nhiên, máy đổi tiền chỉ hỗ trợ việc đổi tiền trọn gói theo qui tắc đổi gói (Z_1, S_1, M_1) lấy gói (Z_2, S_2, M_2) đồng tiền. Các qui tắc đổi tiền khác nhau từng đôi một, được gán số hiệu tuần tự 1,2,3,... và được công bố trước. Hội viên có thể tạo gói tiền thích hợp từ bộ

suu tập của mình để thực hiện việc đổi tiền. Số lần đổi tiền là không hạn chế, tuy nhiên, để ngăn chặn việc đầu cơ, Hiệp hội quy định mỗi hội viên không được giữ quá 4 đồng tiền mỗi loại. Các đồng tiền nhận được sau mỗi lần đổi được gộp lại với các đồng tiền mà hội viên đang có để thành *một bộ suu tập mới* và có thể được sử dụng để đổi trong những lần sau nếu cần.

Yêu cầu: Cho biết số lượng Z, S, M các đồng tiền vàng, bạc, đồng mà Alibaba có ban đầu và các qui tắc đổi tiền. Hãy chỉ ra *số lần đổi tiền ít nhất* để Alibaba có được bộ suu tập có giá trị. Dữ liệu vào đảm bảo luôn có phương án.

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

- + Dòng đầu ghi 6 số nguyên không âm Z, S, M, Z₀, S₀, M₀ ($0 \leq Z, S, M, Z_0, S_0, M_0 \leq 4$);
- + Các dòng tiếp theo mỗi dòng ghi 6 số nguyên không âm Z₁, S₁, M₁, Z₂, S₂, M₂ xác định một qui tắc đổi tiền.

Kết quả: Đưa ra file văn bản COLLECT.OUT một dòng ghi kết quả tìm được.

Các số trên một dòng của file dữ liệu và kết quả đặt cách nhau ít nhất một dấu cách.

Ví dụ:

COLLECT.INP	COLLECT.OUT
4 0 1 3 3 3	2 GT: Sử dụng 2 qui tắc đổi tiền là 3, 4
1 0 1 0 2 2	
0 1 1 0 0 3	
2 0 1 1 2 3	
1 0 0 1 1 0	

Độc đĩa (Đề thi chọn HSG Quốc gia Bài 3/ 13/3/2002).

Các kỹ sư của một công ty tin học đang thử nghiệm chế tạo đĩa từ có dung lượng thông tin cực lớn. Đĩa có nhiều đường ghi và khoảng cách giữa 2 đường ghi liên tiếp nhau là rất nhỏ. Các đường ghi được đánh số từ 0 đến N, từ ngoài vào trong. Đối với loại đĩa này, việc dịch chuyển đầu đọc từ một đường ghi sang đường ghi kế tiếp là rất khó khăn vì khó đảm bảo độ chính xác cao cho các chuyển động cơ học trên khoảng cách quá bé do không có đủ thời gian để khởi động và phanh đầu đọc. Người ta thiết kế mạch điều khiển với hai lệnh: Lệnh T và Lệnh L.

Lệnh T đưa đầu đọc tiến lên phía trước P đường ghi ($P > 0$). Ví dụ đầu đọc đang ở đường ghi K, sau khi thực hiện lệnh T thì nó chuyển tới đường ghi số K+P. Lệnh T không áp dụng được khi $K+P > N$.

Lệnh L đưa đầu đọc lùi Q đường ghi ($Q > 0$). Ví dụ đầu đọc đang ở đường ghi K, sau khi thực hiện lệnh L thì đầu đọc chuyển tới đường ghi số K-Q. Lệnh L không áp dụng được khi $K-Q < 0$.

Để di chuyển đầu đọc từ đường ghi U tới đường ghi V có thể phải áp dụng một dãy các lệnh T (L) liên tiếp nhau được viết gọn dưới dạng $T_m (L_m)$, trong đó m - số nguyên dương, $m \geq 1$.

Yêu cầu: Với N, P và Q cho trước ($0 < N \leq 20\,000$, $0 < P, Q < N$) hãy chỉ ra dãy ít nhất câu lệnh L, T đưa đầu đọc từ đường ghi U tới đường ghi V ($0 \leq U, V \leq N$) hoặc cho biết không tồn tại dãy câu lệnh như vậy.

Ví dụ: Với $N = 10$, $P = 5$, $Q = 3$, $U = 7$, $V = 6$, dãy các lệnh chuyển đầu đọc có thể là L2 T1. Số lệnh phải thực hiện là 3.

Dữ liệu: Vào từ file văn bản DISK.INP gồm 1 dòng 5 số nguyên N P Q U V, các số trên một dòng cách nhau ít nhất một dấu cách.

Kết quả: Đưa ra file văn bản DISK.OUT

- ❖ Dòng đầu tiên là số nguyên K - Số lệnh cần thực hiện, $K = -1$ nếu không tồn tại cách đưa đầu đọc về đường ghi V.
- ❖ Dòng thứ hai chứa dãy câu lệnh cần thực hiện, trước tên lệnh T (L) phải có một dấu cách.

Ví dụ:

DISK.INP	DISK.OUT
10 5 3 7 6	3
	L2 T1

Siêu mã (SIEUMA.cpp)

Trên bàn cờ kích thước vô hạn có một quân siêu mã. Quân cờ này có thể đi theo một số kiểu khác nhau. Mỗi nước đi được mô tả bằng 2 số nguyên: số thứ nhất cho biết quân đó có thể vượt qua bao nhiêu cột (sang phải nếu số này dương và sang trái nếu số này âm), số thứ hai cho biết số dòng vượt qua (lên trên nếu số này dương và xuống dưới nếu số này âm) khi thực hiện nước đi.

Yêu cầu: Hãy xác định xem quân siêu mã này, xuất phát từ một ô cố định nào đó, có thể tới ô bất kỳ trên bàn cờ được không?

Dữ liệu : Vào từ file SIEUMA.INP gồm:

- + Dòng đầu tiên chứa số nguyên N - số nước đi khác nhau của siêu mã ($1 \leq N \leq 100$).
- + Mỗi dòng trong N dòng tiếp theo chứa 2 số nguyên P, Q mô tả một nước đi của quân siêu mã. ($-100 \leq P, Q \leq 100$).

Kết quả: Ghi ra file văn bản SIEUMA.OUT là câu trả lời YES hoặc NO.

Ghi YES nếu quân siêu mã xuất phát từ một ô cố định nào đó, có thể tới được ô bất kỳ trên bàn cờ. Ngược lại, ghi NO.

Ví dụ:

SIEUMA.INP	SIEUMA.OUT
5 3 4 -3 -6 2 -2 5 6 -1 4	NO
SIEUMA.INP	SIEUMA.OUT
5 2 2 -1 -1 2 -2 -1 1 -5 -4	YES

Đổi thẻ (Doithe.cpp)

Có N loại thẻ (đánh số từ 1 đến N). Một máy đổi thẻ có N cửa (đánh số từ 1 đến N) hoạt động theo nguyên tắc sau: nếu đưa thẻ có số hiệu i vào cửa số j thì máy sẽ trả lại thẻ có số hiệu $A[i, j] \in \{1, 2, 3, \dots, N\}$. Nếu một người có một thẻ nào đó, muốn đổi lấy một thẻ có số hiệu mong muốn thì phải đổi nhiều lần (mỗi lần mang thẻ hiện có vào đổi tại một cửa thích hợp).

Yêu cầu:

1. Từ thẻ ban đầu, đổi lấy thẻ mong muốn.
2. Từ thẻ ban đầu, đổi lấy thẻ có số hiệu lớn nhất có thể được.

Trong các trường hợp, đều cần tìm phương án có số lần đổi thẻ ít nhất.

Dữ liệu: Vào từ file văn bản DOITHE.INP gồm các dòng:

- ❖ Dòng đầu ghi giá trị N .

❖ Các dòng sau, dòng i ghi N số hiệu $A[i, 1], A[i, 2], \dots, A[i, N]$.
($i = 1, 2, 3, \dots, N$).

❖ Dòng cuối ghi số hiệu thẻ ban đầu s và số hiệu thẻ mong muốn t .

Kết quả: Ghi ra file DOITHE.OUT gồm các dòng:

❖ Dòng đầu ghi số K là số lần đổi thẻ để được thẻ mong muốn, nếu không thẻ đổi được thì ghi -1, nếu không cần đổi (vì thẻ ban đầu chính là thẻ mong muốn) thì ghi 0.

❖ Dòng sau ghi K số hiệu cửa theo thứ tự cần đến để đổi thẻ, nếu $K < 1$ thì dòng này bỏ qua.

❖ Dòng tiếp theo ghi số M là số lần đổi thẻ để được thẻ có số hiệu lớn nhất, nếu thẻ ban đầu đồng thời là thẻ có số hiệu lớn nhất thì ghi 0.

❖ Dòng cuối cùng ghi M số hiệu cửa theo thứ tự cần đến để đổi thẻ, nếu $M = 0$ thì dòng này bỏ qua.

Các số trên cùng một dòng trong các file được ghi cách nhau ít nhất một dấu trắng. *Giới hạn kích thước:* $N \leq 200$.

Ví dụ:

DOITHE.INP	DOITHE.OUT
5	3
3 2 4 3 5	1 3 4
3 2 2 2 3	4
3 3 4 3 2	1 3 4 5
4 3 2 1 2	
1 2 3 4 5	
2 1	