

HÀNG ĐỢI

DSA08001.

Ban đầu cho một queue rỗng. Bạn cần thực hiện các truy vấn sau:

1. Trả về kích thước của queue
2. Kiểm tra xem queue có rỗng không, nếu có in ra “YES”, nếu không in ra “NO”.
3. Cho một số nguyên và đẩy số nguyên này vào cuối queue.
4. Loại bỏ phần tử ở đầu queue nếu queue không rỗng, nếu rỗng không cần thực hiện.
5. Trả về phần tử ở đầu queue, nếu queue rỗng in ra -1.
6. Trả về phần tử ở cuối queue, nếu queue rỗng in ra -1.

Dữ liệu vào

Dòng đầu tiên chứa số nguyên T là số bộ dữ liệu, mỗi bộ dữ liệu theo dạng sau.

Dòng đầu tiên chứa số nguyên n - lượng truy vấn ($1 \leq n \leq 1000$)

N dòng tiếp theo, mỗi dòng sẽ ghi loại truy vấn như trên, với truy vấn loại 3 sẽ có thêm một số nguyên, không quá 10^6 .

Kết quả: In ra kết quả của các truy vấn..

Ví dụ:

Input	Output
1	
14	
3 1	1
3 2	3
3 3	5
5	2
6	
4	
4	

4	
4	
4	
3 5	
3 6	
5	
1	

DSA08002.

Yêu cầu bạn xây dựng một queue với các truy vấn sau đây:

“PUSH x”: Thêm phần tử x vào cuối của queue ($0 \leq x \leq 1000$).

“PRINTFRONT”: In ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

“POP”: Xóa phần tử ở đầu của queue. Nếu queue rỗng, không làm gì cả.

Dữ liệu vào:

Dòng đầu tiên là số lượng truy vấn Q ($Q \leq 100000$).

Mỗi truy vấn có dạng như trên.

Kết quả:

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

Ví dụ:

Input	Output
9	2
PUSH 1	2
PUSH 2	NONE

POP	
PRINTFRONT	
PUSH 3	
PRINTFRONT	
POP	
POP	
PRINTFRONT	

DSA08003.

Yêu cầu bạn xây dựng một hàng đợi hai đầu với các truy vấn sau đây:

“PUSHFRONT x”: Thêm phần tử x vào đầu của dequeue ($0 \leq x \leq 1000$).

“PRINTFRONT”: In ra phần tử đầu tiên của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPFRONT”: Xóa phần tử đầu của dequeue. Nếu dequeue rỗng, không làm gì cả.

“PUSHBACK x”: Thêm phần tử x vào cuối của dequeue ($0 \leq x \leq 1000$).

“PRINTBACK”: In ra phần tử cuối của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPBKACK”: Xóa phần tử cuối của dequeue. Nếu dequeue rỗng, không làm gì cả.

Dữ liệu vào:

Dòng đầu tiên là số lượng truy vấn Q ($Q \leq 100000$).

Mỗi truy vấn có dạng như trên.

Kết quả:

Với mỗi truy vấn “PRINTFRONT” và “PRINTBACK”, hãy in ra kết quả trên một dòng.

Ví dụ:

Input	Output
10	2

PUSHBACK 1	1
PUSHFRONT 2	3
PUSHBACK 3	NONE
PRINTFRONT	
POPFRONT	
PRINTFRONT	
POPFRONT	
PRINTBACK	
POPFRONT	
PRINTBACK	

DSA08005.

Cho số tự nhiên n . Hãy in ra tất cả các số nhị phân từ 1 đến n .

Input:

- Dòng đầu tiên ghi lại số lượng test T ($T \leq 100$).
- Mỗi test là một số tự nhiên n được ghi trên một dòng ($n \leq 10000$).

Output:

- Đưa ra kết quả mỗi test trên một dòng.

Ví dụ:

Input	Output
2	1 10
2	1 10 11 100 101
5	

DSA08006.

Cho số tự nhiên N . Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 9 và số 0 chia hết cho N . Ví dụ với $N = 5$ ta sẽ tìm ra $X = 90$.

Input:

- Dòng đầu tiên ghi lại số lượng test T ($T \leq 100$).
- Những dòng kế tiếp mỗi dòng ghi lại một test. Mỗi test là một số tự nhiên N được ghi trên một dòng ($N \leq 100$).

Output:

- Đưa ra theo từng dòng số X nhỏ nhất chia hết cho N tìm được .

Ví dụ:

Input	Output
2	90
5	9009
7	

DS08007.

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số $K = 1, 10, 101$. Cho số tự nhiên N ($N < 2^{63}$). Hãy cho biết có bao nhiêu số BDN nhỏ hơn N . Ví dụ $N=100$ ta có 4 số BDN bao gồm các số: 1, 10, 11, 100.

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
- T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
3	2

10	4
100	7
200	

DSA08008.

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số $K = 101$ là số BDN, $k=102$ không phải là số BDN.

Số BDN của N là số $P = M'N$ sao cho P là số BDN. Cho số tự nhiên N ($N < 1000$), hãy tìm số BDN nhỏ nhất của N .

Ví dụ. Với $N=2$, ta tìm được số BDN của N là $P = 5'2=10$. $N = 17$ ta tìm được số BDN của 17 là $P = 653'17=11101$.

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
- T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N .

Output:

Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
3	10
2	11100
12	11101
17	

DSA08017. Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N . Hãy liệt kê các số lộc phát có không quá N chữ số.

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 10$);
- T dòng kế tiếp mỗi dòng ghi số N ($1 < N < 15$).

Output:

- In ra đáp án theo thứ tự giảm dần.

Ví dụ:

Input	Output
2	88 86 68 66 8 6
2	888 886 868 866 688 686 668 666 88 86 68 66 8 6
3	

DSA08018.

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 10$);
- T dòng kế tiếp mỗi dòng ghi số N ($1 < N < 15$).

Output:

- Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in đáp án **theo thứ tự tăng dần**.

Ví dụ:

Input	Output
2	6
2	6 8 66 68 86 88
3	14 6 8 66 68 86 88 666 668 686 688 866 868 886 888

DSA08019.

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N. Hãy liệt kê các số lộc phát có không quá N chữ số.

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 10$);
- T dòng kế tiếp mỗi dòng ghi số N ($1 < N < 15$).

Output:

- Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in ra đáp án **theo thứ tự giảm dần**.

Ví dụ:

Input	Output
2	6
2	88 86 68 66 8 6
3	14
	888 886 868 866 688 686 668 666 88 86 68 66 8 6

DSA08004.

Cho xâu ký tự S[] bao gồm các ký tự in hoa [A, B, ..., Z]. Ta định nghĩa giá trị của xâu S[] là tổng bình phương số lần xuất hiện mỗi ký tự trong xâu. Ví dụ với xâu S[] = "AAABBCD" ta có $F(S) = 3^2 + 2^2 + 1^2 + 1^2 = 15$. Hãy tìm giá trị nhỏ nhất của xâu S[] sau khi loại bỏ K ký tự trong xâu.

Input:

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$).
- Mỗi test được tổ chức thành 2 dòng. Dòng thứ nhất ghi lại số K. Dòng thứ 2 ghi lại xâu ký tự S[] có độ dài không vượt quá 10^6 .

Output:

- Đưa ra giá trị nhỏ nhất của mỗi test theo từng dòng.

Input	Output
-------	--------

2	
0	
ABCC	6
1	3
ABCC	

DSA08009.

Cho hai số nguyên dương S và T ($S, T < 10000$) và hai thao tác (a), (b) dưới đây:

Thao tác (a): Trừ S đi 1 ($S = S - 1$);

Thao tác (b): Nhân S với 2 ($S = S * 2$);

Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất. Ví dụ với $S = 2, T = 5$ thì số các bước ít nhất để dịch chuyển S thành T thông qua 4 thao tác sau:

Thao tác (a): $2 * 2 = 4$;

Thao tác (b): $4 - 1 = 3$;

Thao tác (a): $3 * 2 = 6$;

Thao tác (b): $6 - 1 = 5$;

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
- T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một bộ đôi S và T.

Output: Đưa ra kết quả mỗi test theo từng dòng.

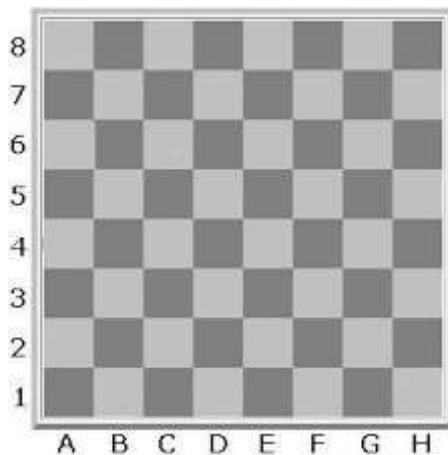
Ví dụ:

Input	Output
3	4
2 5	4
	3

3 7	
7 4	

DSA08025.

Cho một quân mã trên bàn cờ vua tại vị trí ST. Nhiệm vụ của bạn là hãy tìm số bước di chuyển ít nhất để đưa quân mã tới vị trí EN.



Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm 2 xâu dạng “xy” và “uv”, trong đó x, y là kí tự trong “abcdefgh” còn y, v là số thuộc 1, 2, 3, 4, 5, 6, 7, 8.

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

Ví dụ:

Input:	Output
8	2
e2 e4	4
a1 b2	2

b2 c3	6
a1 h8	5
a1 h7	6
h8 a1	1
b1 c3	0
f6 f6	

DSA08010.

Cho số tự nhiên N ($N < 10^9$) và hai phép biến đổi (a), (b) dưới đây.

- **Thao tác (a):** Trừ N đi 1 ($N = N - 1$). Ví dụ $N = 17$, thao tác (a) biến đổi $N = N - 1 = 16$.
- **Thao tác (b):** $N = \max(u, v)$ nếu $u * v = N$ ($u > 1, v > 1$). Ví dụ $N = 16$, thao tác (b) có thể biến đổi $N = \max(2, 8) = 8$ hoặc $N = \max(4, 4) = 4$.

Chỉ được phép sử dụng hai thao tác (a) hoặc (b), hãy biến đổi N thành 1 sao số các thao tác (a), (b) được thực hiện ít nhất. Ví dụ với $N = 17$, số các phép (a), (b) nhỏ nhất biến đổi N thành 1 là 4 bước như sau:

$$\text{Thao tác (a): } N = N - 1 = 17 - 1 = 16$$

$$\text{Thao tác (b): } 16 = \max(4, 4) = 4$$

$$\text{Thao tác (b): } 4 = \max(2, 2) = 2$$

$$\text{Thao tác (a): } 2 = 2 - 1 = 1$$

Input:

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
- T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số N .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
3	4
17	5
50	5
100	

DSA08012.

Cho hai số nguyên dương L, R . Hãy đưa ra số các số K trong khoảng $[L, R]$ thỏa mãn điều kiện:

- Tất cả các chữ số của K đều khác nhau.
- Tất cả các chữ số của K đều nhỏ hơn hoặc bằng 5.

Ví dụ với $L = 4, R = 13$ ta có 5 số thỏa mãn yêu cầu là 4, 5, 10, 12, 13,

Input:

- Dòng đầu tiên đưa vào số lượng test T .
- Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một cặp L, R được viết trên một dòng.
- T, L, R thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq L \leq R \leq 10^5$.

Output:

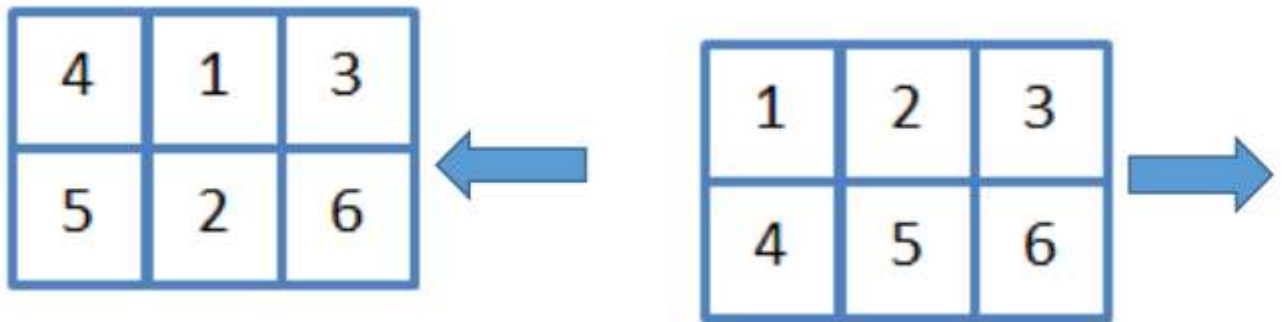
- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
2	5
4 13	100
100 1000	

DSA08029.

Có một chiếc bảng hình chữ nhật với 6 miếng ghép, trên mỗi miếng ghép được điền một số nguyên trong khoảng từ 1 đến 6. Tại mỗi bước, chọn một hình vuông (bên trái hoặc bên phải), rồi quay theo chiều kim đồng hồ.



Yêu cầu: Cho một trạng thái của bảng, hãy tính số phép biến đổi ít nhất để đưa bảng đến trạng thái đích.

Input:

- Dòng đầu ghi số bộ test (không quá 10). Mỗi bộ test gồm hai dòng:
 - Dòng đầu tiên chứa 6 số là trạng thái bảng ban đầu (thứ tự từ trái qua phải, dòng 1 tới dòng 2).
 - Dòng thứ hai chứa 6 số là trạng thái bảng đích (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Output:

- Với mỗi test, in ra một số nguyên là đáp số của bài toán.

Ví dụ:

Input	Output
1	
1 2 3 4 5 6	
4 1 2 6 5 3	2

DSA08020.

Cho cặp số S và T là các số nguyên tố có 4 chữ số (Ví dụ S = 1033, T = 8197 là các số nguyên tố có 4 chữ số). Hãy viết chương trình tìm cách dịch chuyển S thành T thỏa mãn đồng thời những điều kiện dưới đây:

- Mỗi phép dịch chuyển chỉ được phép thay đổi một chữ số của số ở bước trước đó (ví dụ nếu S=1033 thì phép dịch chuyển S thành 1733 là hợp lệ);

- b. Số nhận được cũng là một số nguyên tố có 4 chữ số (ví dụ nếu $S=1033$ thì phép dịch chuyển S thành 1833 là không hợp lệ, và S dịch chuyển thành 1733 là hợp lệ);
- c. Số các bước dịch chuyển là ít nhất.

Ví dụ số các phép dịch chuyển ít nhất để $S = 1033$ thành $T = 8179$ là 6 bao gồm các phép dịch chuyển như sau:

8179 B 8779B 3779B 3739B 3733B 1733B 1033.

Input:

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$)
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là một bộ đôi S, T .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
2	6
1033 8179	5
1033 8779	

DSA08022.

Cho dãy số $A[]$ có N phần tử là các chữ số từ 0 đến 9. Nhiệm vụ của bạn là hãy chọn lấy một tổ hợp các phần tử và sắp xếp chúng sao cho thu được số lớn nhất chia hết cho 3.

Nếu không tìm được số nào, in ra -1.

Input:

- Dòng đầu tiên là số lượng bộ test T ($1 \leq T \leq 50$).
- Mỗi test bắt đầu bởi số nguyên N ($1 \leq N \leq 100\,000$).
- Dòng tiếp theo gồm N số nguyên $A[i]$ ($0 \leq A[i] \leq 9$).

Output:

- Với mỗi test, in ra đáp án tìm được trên một dòng.

Ví dụ:

Input	Output
3	981
3	8760
8 1 9	-1
5	
8 1 7 6 0	
2	
5 2	

DSA08011.

Cho tập n chuỗi ký tự S và hai chuỗi s, t thuộc S . Ta giả thiết các chuỗi ký tự $S[i]$ thuộc S có độ dài bằng nhau. Hãy tìm khoảng cách đường đi ngắn nhất từ s đến t . Biết từ một chuỗi ký tự bất kỳ ta chỉ được phép dịch chuyển đến chuỗi khác với nó duy nhất 1 ký tự. Ví dụ ta có tập các từ $S = \{ \text{POON, TOON, PLEE, SAME, POIE, PLEA, PLIE, POIN} \}$, $s = \text{TOON}$, $t = \text{PLEA}$ ta có độ dài đường đi ngắn nhất là 7 tương ứng với các phép dịch chuyển : $\text{TOON} \rightarrow \text{POON} \rightarrow \text{POIN} \rightarrow \text{POIE} \rightarrow \text{PLIE} \rightarrow \text{PLEE} \rightarrow \text{PLEA}$.

Input:

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$).
- Mỗi test được tổ chức thành 2 dòng. Dòng thứ nhất ghi lại n là số từ trong S và hai từ s, t . Dòng thứ 2 đưa vào n chuỗi ký tự của S ; các chuỗi ký tự được viết cách nhau một vài khoảng trống, có độ dài không vượt quá 10 ký tự.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

Input	Output
1	7

8 TOON PLEA	
POON TOON PLEE SAME POIE PLEA PLIE POIN	

DSA08013.

Cho một bảng kích thước $N \times N$, trong đó có các ô trống '.' và vật cản 'X'. Các hàng và các cột được đánh số từ 0.

Mỗi bước di chuyển, bạn có thể đi từ ô (x, y) tới ô (u, v) nếu như 2 ô này nằm trên cùng một hàng hoặc một cột, và không có vật cản nào ở giữa.

Cho điểm xuất phát và điểm đích. Bạn hãy tính số bước di chuyển ít nhất?

Input:

- Dòng đầu ghi số bộ test (không quá 10). Mỗi test gồm:
 - Dòng đầu tiên là số nguyên dương N ($1 \leq N \leq 100$).
 - N dòng tiếp theo, mỗi dòng gồm N kí tự mô tả bảng.
 - Cuối cùng là 4 số nguyên a, b, c, d với (a, b) là tọa độ điểm xuất phát, (c, d) là tọa độ đích. Dữ liệu đảm bảo hai vị trí này không phải là ô có vật cản.

Output:

- Với mỗi test, in ra một số nguyên là đáp số của bài toán.

Ví dụ:

Input	Output
1	
3	
.X.	3
.X.	
...	

0 0 0 2	
---------	--

DSA08014.

Trên một giá có kích thước $R \times C$ (R hàng, C cột), một số hạt mầm đã được tra vào các ô. Một số hạt mầm được bón thêm chất dinh dưỡng, nên đã nảy mầm sớm thành cây non.

Mỗi ngày, các cây non sẽ lan truyền chất dinh dưỡng của nó cho các mầm ở ô xung quanh (trái, trên, phải, dưới), làm cho các hạt mầm này phát triển thành cây non. Tuy nhiên, có thể có một số hạt mầm được gieo ở vị trí lẻ loi, do không nhận được chất dinh dưỡng nên không thể nảy mầm.

Các bạn hãy xác định xem cần ít nhất bao nhiêu ngày để tất cả các hạt đều mầm?

Input:

- Dòng đầu ghi số bộ test (không quá 10). Mỗi bộ test gồm:
 - Dòng đầu tiên gồm 2 số nguyên R và C ($1 \leq R, C \leq 500$).
 - R dòng tiếp theo, mỗi dòng gồm C số nguyên $A[i][j]$.
 - $A[i][j] = 0$, ô (i, j) là ô trống.
 - $A[i][j] = 1$, ô (i, j) là hạt chưa nảy mầm.
 - $A[i][j] = 2$, ô (i, j) là cây non.

Output:

- Với mỗi test in ra thời gian ngắn nhất để tất cả các hạt đều nảy mầm. Nếu có hạt nào chưa nảy mầm, in ra -1.

Ví dụ:

Input	Output
2	
3 5	2
2 1 0 2 1	-1
1 0 1 2 1	
1 0 0 2 1	

3 5	
2 1 0 2 1	
0 0 1 2 1	
1 0 0 2 1	

DSA08015.

Cho một hình hộp chữ nhật có kích thước $A \times B \times C$, trong đó A là chiều cao, B là chiều rộng và C là chiều dài. Mỗi ô có thể là một ô trống '.' hoặc vật cản '#'.
Mỗi bước, bạn được phép di chuyển sang một ô kề bên cạnh (không được đi chéo).
Nhiệm vụ của bạn là tìm đường đi ngắn nhất bắt đầu 'S' tới vị trí kết thúc 'E'.

Input:

- Dòng đầu tiên là số lượng bộ test T ($1 \leq T \leq 50$).
- Mỗi test bắt đầu bởi 3 số nguyên A, B, C ($A, B, C \leq 30$).
- Tiếp theo là A khối, mỗi khối gồm $B \times C$ kí tự mô tả một lát cắt của hình hộp chữ nhật. Giữa 2 khối có một dấu xuống dòng.

Output:

- In ra một số nguyên là đường đi ngắn nhất từ S tới E. Nếu không di chuyển được, in ra -1.

Ví dụ:

Input	Output
2	
3 4 5	
S....	11
.###.	-1
.##..	
###.#	

#####	
#####	
##.##	
##...	
#####	
#####	
###.###	
#####E	
1 3 3	
S##	
#E#	
###	

DSA08045.

Trên cánh đồng có N hàng rào. Mỗi hàng rào được miêu tả bằng một đoạn thẳng đứng hoặc ngang trong mặt phẳng hai chiều. Nếu hai hàng rào gặp nhau, chúng chỉ có thể gặp nhau ở đầu mút của đoạn thẳng. Mỗi hàng rào chỉ giao với hai hàng rào khác tại các kiểu đầu mút của đoạn thẳng.

Có M chú bò đang ăn cỏ trên cánh đồng. Mỗi chú bò được thể hiện bằng một điểm trên mặt phẳng tọa độ hai chiều. Input đảm bảo không có chú bò nào nằm trên đoạn thẳng đại diện cho các hàng rào, và không có hai chú bò có cùng tọa độ. Các chú bò có thể di chuyển tự do trong vùng bao quanh bởi các hàng rào và tạo thành bầy đàn.

Các bạn hãy tìm xem đàn bò nào có số lượng nhiều nhất?

Input: Dòng đầu ghi số bộ test (không quá 10). Mỗi bộ test như sau:

- Dòng đầu tiên là 2 số N, M ($1 \leq N, M \leq 500$).
- N dòng tiếp theo, mỗi dòng gồm 4 số nguyên A_x, A_y, B_x, B_y mô tả tọa độ của một hàng rào.
- M dòng tiếp theo, mỗi dòng gồm 2 số nguyên C_x, C_y mô tả tọa độ của một chú bò.
- Các tọa độ có giá trị trong khoảng từ 0 tới 10^6 .

Output:

- In ra số lượng lớn nhất có thể của một đàn bò.

Ví dụ:

Input:	Output:
1	
7 3	
0 0 10 0	
10 0 10 5	
12 5 10 5	
10 5 1 5	
12 5 12 7	2
0 7 12 7	
0 7 0 0	
3 4	
6 6	
17 3	

DSA08016.

HEXGAME là một trò chơi xếp hình gồm 10 miếng ghép hình lục giác đều, trên mỗi miếng ghép được điền một số nguyên, có 8 miếng được điền số từ 1 đến 8 và có hai miếng điền số 0. Các miếng liên kết với nhau tạo thành lưới tổ ong. Ban đầu các miếng

ghép ở vị trí như hình vẽ. Tại mỗi bước, chọn một miếng ghép có đúng 6 miếng ghép kề cạnh làm tâm, rồi xoay một nấc 6 miếng ghép kề cạnh đó theo chiều kim đồng hồ. Như vậy chỉ có hai cách chọn tâm, đó là chọn tâm bên trái và chọn tâm bên phải.

Yêu cầu: Cho một trạng thái của trò chơi (nhận được sau một dãy biến đổi từ trạng thái ban đầu), hãy tính số phép biến đổi ít nhất để đưa về trạng thái ban đầu.

Input:

- Dòng đầu ghi số bộ test (không quá 10). Mỗi bộ test gồm:
 - Dòng đầu tiên chứa 3 số ở 3 miếng ghép dòng thứ nhất (thứ tự từ trái qua phải).
 - Dòng thứ hai chứa 4 số ở 4 miếng ghép dòng thứ hai (thứ tự từ trái qua phải).
 - Dòng thứ 3 chứa 3 số ở 3 miếng ghép dòng thứ ba (thứ tự từ trái qua phải).

Output:

- Với mỗi bộ test in ra một số nguyên là số phép biến đổi ít nhất để đưa được về trạng thái ban đầu.

Ví dụ:

Input	Output
1	5
1 0 2	
8 6 0 3	
7 5 4	