

## snakegame

Trên bảng kích thước  $m \times n$ , máy tính chọn ngẫu nhiên một số ô là hang rắn được kí hiệu bởi kí tự +. Người chơi cần di chuyển từ vị trí là ô chứa kí tự P đến ô chứa kí tự C. Một cách di chuyển được gọi là thông minh nếu càng tránh xa các ô là hang rắn càng tốt. Khoảng cách hai ô  $(x, y)$  và  $(u, v)$  được tính theo khoảng cách Manhattan:  $|x - u| + |y - v|$ .

**Yêu cầu:** Tìm cách di chuyển để trong quá trình di chuyển khoảng cách tới ô là hang rắn ngắn nhất là xa nhất.

### Input

- Dòng đầu chứa hai số nguyên  $m, n$  ( $m, n \leq 2000$ );
- Tiếp theo là  $m$  hàng, mỗi hàng chứa  $n$  kí tự là một trong các kí tự: '+' (hang rắn), 'P' (xuất phát), 'C' (đích), '.' (ô tự do).

### Output

- Gồm một dòng chứa một số là khoảng cách tìm được.

Dữ liệu vào	Kết quả ra
4 5 P.... ..... +++.. ....C	2

**Subtask 1:**  $m, n \leq 50$ ;

**Subtask 2:** Không có ràng buộc nào thêm.

game44

Cho bảng số  $4 \times 4$ , mỗi ô chứa một số nguyên từ 0 đến 15, các số đôi một khác nhau. Ô chứa số 0 được hiểu là ô trống, mỗi bước được đẩy một ô chứa số khác 0 vào ô trống.

A

1	3	5	6
4	2		10
12	7	14	11
13	9	15	8

B

1	3	5	6
4	7	2	10
12		14	11
13	9	15	8

C

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

**Yêu cầu:** Tìm cách đẩy ít bước nhất để nhận được bảng chuẩn (như hình C), biết rằng tồn tại một cách biến đổi không quá 25 bước.

Input

- Gồm 4 dòng, mỗi dòng chứa 4 số mô tả bảng số ban đầu.

Output

- Đưa ra số bước ít nhất cần đẩy để nhận được bảng chuẩn.

Dữ liệu vào	Kết quả ra
1 2 3 4 5 0 6 8 9 10 7 11 13 14 15 12	4 6 7 11 12

## Kfib

Dãy Fibonacci được định nghĩa như sau:

$$f(i) = \begin{cases} 0 & \text{nếu } i = 0 \\ 1 & \text{nếu } i = 1 \\ f(i-1) + f(i-2) & \text{nếu } i > 1 \end{cases}$$

Cho  $M, A, B$  tìm  $k$  nhỏ nhất mà  $f(k) \% M = A$  và  $f(k+1) \% M = B$ .

### Input

- Dòng đầu đầu là số  $T$  là số bộ dữ liệu ( $T \leq 3$ );
- Tiếp theo là  $T$  dòng, mỗi dòng chứa 3 số nguyên  $M, A, B$  ( $0 \leq A, B < M; M > 0$ );

### Output

- Gồm  $T$  dòng, mỗi dòng là kết quả cho bộ dữ liệu tương ứng, nếu không tồn tại  $k$  ghi -1.

Dữ liệu vào	Kết quả ra
2	0
10 0 1	5
3 2 2	

**Subtask 1:**  $M \leq 10^6$ ;

**Subtask 2:**  $M \leq 10^9$ ;

## wbpath

Trò chơi tìm đường trên bảng chữ như sau: Bảng có kích thước  $m \times n$  và chỉ gồm hai loại kí tự  $w$  và  $b$ . Các hàng của bảng được đánh số từ 1 đến  $m$  từ trên xuống, các cột được đánh số từ 1 đến  $n$  từ trái sang, ô nằm giao giữa hàng  $i$  và cột  $j$  gọi là ô  $(i, j)$ . Nhiệm vụ của người chơi là tìm đường đi từ ô  $(x_s, y_s)$  tới ô  $(x_t, y_t)$  mà lượng số kí tự  $w$  và  $b$  bằng nhau. Số lượng kí tự  $w, b$  trên đường đi là số lượng kí tự nằm trong các ô đi qua, nếu đường đi đi qua một ô nhiều lần thì kí tự đó cũng được tính nhiều lần.

**Yêu cầu:** Cho bảng chữ và  $q$  nhiệm vụ, với mỗi nhiệm vụ cần tìm đường đi hoặc đưa ra  $-1$  nếu không tồn tại đường đi thỏa mãn.

### Input

- Dòng đầu chứa hai số  $m, n$  ( $m, n \leq 1000$ );
- Dòng thứ  $i$  ( $1 \leq i \leq m$ ) trong  $m$  dòng tiếp theo chứa một xâu độ dài  $n$ ;
- Dòng tiếp theo chứa số nguyên  $q$  ( $q \leq 10$ );
- Dòng thứ  $k$  ( $1 \leq k \leq q$ ) trong  $q$  dòng tiếp theo chứa bốn số  $x_s, y_s, x_t, y_t$ .

### Output

- Với mỗi nhiệm vụ ghi ra trên một dòng, đưa ra  $-1$  nếu không tồn tại đường đi thỏa mãn, ngược lại đưa ra theo khuôn dạng: Số nguyên  $d$  là số ô lần lượt được đi, tiếp theo là một dấu cách, cuối cùng là xâu  $p$  độ dài  $d - 1$  chỉ gồm các kí tự  $U, D, L, R$  tương ứng việc di chuyển lên trên, xuống dưới, sang trái, sang phải.

Dữ liệu vào	Kết quả ra
3 3	4 DRU
WWB	-1
BBB	-1
BBW	4 RDD
4	
1 1 1 2	
1 1 1 1	
1 1 3 3	
1 1 3 2	

**Subtask 1:**  $m, n \leq 5$ ;

**Subtask 2:**  $m, n \leq 25$ ;

**Subtask 3:**  $m, n \leq 100$ ;

**Subtask 4:**  $m, n \leq 1000$ ;

## spnum

Một số nguyên dương gọi là số spnum nếu tổng các chữ số bằng  $s$  và tổng bình phương các chữ số bằng  $p$ .

**Yêu cầu:** Cho hai số nguyên dương  $s, p$ . Hãy tìm số spnum nhỏ nhất thỏa mãn.

### Input

- Dòng đầu chứa số bộ dữ liệu  $T$ ;
- Tiếp theo là  $T$  dòng, mỗi dòng chứa hai số nguyên dương  $s, p$ .

### Output

- Gồm  $T$  dòng, mỗi dòng là kết quả tương ứng với dữ liệu vào. Nếu không tồn tại số spnum hoặc số spnum nhỏ nhất tìm được có hơn 100 chữ số thì ghi No solution.

Dữ liệu vào	Kết quả ra
3	No solution
5 6	23
5 13	111
3 3	

**Subtask 1:**  $T \leq 10; s, p \leq 100$ ;

**Subtask 2:**  $T \leq 10^3; s, p \leq 10^4$ ;