

01/08:

Bài A. NUM13

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Cho số nguyên dương n . Hãy đếm xem có bao nhiêu số nguyên dương x chia hết cho 13; sao cho ở dạng biểu diễn thập phân, x có đúng n chữ số, đồng thời x không chứa chữ số 1 và 3.

Dữ liệu vào

Gồm một số nguyên dương n ($1 \leq n \leq 10^5$)

Kết quả

Ghi số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
1	0
2	4

Giải thích

Các số thỏa mãn test VD 2 là 39, 52, 65, 78

Hạn chế

- Có 20% số test với $n \leq 6$

Bài B. RTREE

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Mạng lưới thông tin trên hành tinh XYZ có thể được hiểu như một cây n đỉnh (một đồ thị liên thông không có chu trình), mỗi đỉnh là một trạm, còn mỗi cạnh là một đường truyền. Có m gói tin, gói thứ i cần truyền từ trạm u_i đến trạm v_i

Quân đội trái đất cần phá hủy một đường truyền (1 cạnh trên cây) để số gói tin không thể gửi đi là lớn nhất, bạn hãy lập trình tính toán giúp họ

Dữ liệu vào

- Dòng đầu: n m
- $n - 1$ dòng tiếp theo ghi các cạnh của cây: u v
- m dòng tiếp theo, dòng thứ i ghi: u_i v_i

Kết quả

- Một số nguyên là số gói tin không thể gửi đi

Ví dụ

stdin	stdout
3 2 1 2 1 3 2 3 2 1	2

Hạn chế

- $1 \leq n, m \leq 10^5$
- 50% số test có $1 \leq n, m \leq 5000$

Bài C. MININTERVAL

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho dãy số a có độ dài n và một số k . Đếm số lượng cặp i, j ($1 \leq i \leq j \leq n$) sao cho $a_i + a_j + \min(a_i, a_{i+1}, \dots, a_j) \leq k$.

Dữ liệu vào

- Dòng đầu tiên chứa 2 số n và k ($n \leq 500000, 0 \leq k \leq 10^{13}$);
- Dòng thứ 2 chứa n số nguyên là giá trị của các số trong mảng a ($0 \leq a_i \leq 10^{13}$).

Kết quả

Một số duy nhất là số lượng cặp i, j thỏa mãn điều kiện đề bài.

Ví dụ

stdin	stdout
5 13 12 0 4 10 12	6
5 10 3 9 2 5 3	8

Hạn chế

- Có 12% số test với $n \leq 5000$;
- Có 28% số test với a_i đôi một phân biệt;
- Có 60% số test không có ràng buộc gì thêm;

Bài D. COVID

File dữ liệu vào: `stdin`
File kết quả: `stdout`

Có 1000 người đang xếp hàng nhập cảnh, được đánh số từ 1 đến 1000. Trong số đó có không quá 2 người bị nhiễm covid và tất cả những người bị nhiễm đều đứng cạnh nhau. Có 10 kit test PCR, mỗi kit có thể trộn nhiều mẫu lại để test, kết quả test là dương tính khi và chỉ khi tồn tại một mẫu trong số đó bị nhiễm. Có thể lấy mẫu nhiều lần cho mỗi người, nhưng vì vội nên họ không thể chờ lâu, do đó cần test đồng thời 10 kit (không thể chờ kết quả của kit này rồi mới bắt đầu test kit kia được).

Yêu cầu: Hãy tìm ra một người bị nhiễm hoặc thông báo là không có ai bị nhiễm

Tương tác:

Đầu tiên bạn cần đọc vào số nguyên dương T là số lượng testcase ($1 \leq T \leq 10$). Với mỗi testcase:

- Trước tiên bạn cần in ra 10 dòng, dòng thứ i mô tả kit test thứ i : Gồm số nguyên dương k_i theo sau bởi k_i số nguyên dương phân biệt là chỉ số của những người được lấy mẫu trộn trong kit thứ i
- Tiếp theo bạn cần flush luồng ra chuẩn (`fflush(stdout)` hoặc `cout<<endl`)
- Sau đó bạn cần đọc vào một xâu độ dài 10 là kết quả của các kit, ký tự thứ i là $+/$ - tương ứng là dương tính/âm tính
- Tiếp theo bạn cần in ra một số nguyên là chỉ số của người bị nhiễm. Nếu có 2 người bị nhiễm thì chỉ cần đưa ra người tùy ý trong 2 người đó. Nếu không có ai bị nhiễm thì in ra số 0
- Cuối cùng bạn cần flush luồng ra chuẩn (`fflush(stdout)` hoặc `cout<<endl`)

Thông tin những người bị nhiễm trong input là cố định, không bị thay đổi trong quá trình tương tác

Ví dụ

stdin	stdout
1 --+---	1 1 1 2 1 3 1 4 6 1 2 3 4 5 6 3 2 4 6 0 0 0 0 5

Hạn chế

- Subtask 1 (30%): T chia hết cho 3 và mỗi testcase đều có đúng 1 người bị nhiễm
- Subtask 2 (20%): T chia 3 dư 1 và mỗi testcase đều có đúng 2 người bị nhiễm
- Subtask 3 (50%): T chia 3 dư 2

Bài E. NUM17

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Cho hai số nguyên dương L, R . Hãy đếm xem có bao nhiêu số nguyên dương x thuộc đoạn $[L, R]$ hết cho 17; sao cho ở dạng biểu diễn thập phân, x không chứa hai chữ số bằng nhau nằm kề nhau

Dữ liệu vào

Gồm hai dòng chứa hai số nguyên dương L, R ($1 \leq L \leq R \leq 10^{10000}$)

Kết quả

Ghi số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

<code>stdin</code>	<code>stdout</code>
1 100	5

Giải thích

Các số thỏa mãn là 17, 34, 51, 68, 85

Hạn chế

- Có 20% số test với $R \leq 10^6$

Bài F. RTREE2

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Mạng lưới thông tin trên hành tinh XYZ có thể được hiểu như một cây n đỉnh (một đồ thị liên thông không có chu trình), mỗi đỉnh là một trạm, còn mỗi cạnh là một đường truyền. Các trạm được đánh số từ 1 đến n , các đường truyền được đánh số từ 1 đến $n - 1$.

Mỗi kết nối trên mạng là 1 luồng dữ liệu đang được truyền tải, thể hiện bởi số hiệu của 2 trạm gửi-nhận. Cho m sự kiện theo thứ tự, mỗi sự kiện thuộc 1 trong 2 dạng sau:

- 1 $u\ v$: Có thêm một kết nối giữa u và v
- 2 i : Quân đội muốn tính số kết nối bị gián đoạn nếu cắt cạnh thứ i

Dữ liệu vào

- Dòng đầu: $n\ m$
- $n - 1$ dòng tiếp theo ghi các cạnh của cây: $u\ v$
- m dòng tiếp theo ghi các sự kiện theo mô tả ở trên

Kết quả

- Với mỗi sự kiện loại 2, in ra kết quả trên một dòng. Lưu ý là chúng ta chỉ tính toán chứ chưa thực sự phá hủy đường truyền nào

Ví dụ

stdin	stdout
3 3 1 2 1 3 1 2 3 1 2 1 2 1	2

Hạn chế

- $1 \leq n, m \leq 10^5$
- 50% số test có $1 \leq n, m \leq 5000$

Bài G. LONGESTPATH

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Giải vô địch các phần mềm cờ tướng vừa diễn ra. Theo luật, hai máy bất kỳ đều chơi với nhau phân tiên hậu cho đến khi một trong hai máy dành phần thắng, do đó sẽ không có kết quả hòa. Bảng kết quả có thể mô tả như một ma trận nhị phân: $a_{i,j} = 0/1$ tương ứng là máy i thua/thắng máy j

Một chuỗi trận x_1, x_2, \dots, x_k là đẹp mắt nếu máy x_i thắng máy x_{i+1} với mọi $1 \leq i < k$ và $x_i \neq x_j$ với mọi $i \neq j$. Ban tổ chức cần chọn ra chuỗi trận đẹp mắt dài nhất để trình chiếu cho người hâm mộ xem. Hãy giúp họ tìm một chuỗi trận đẹp mắt dài nhất

Dữ liệu vào

- Dòng đầu chứa số lượng máy: n
- n dòng tiếp theo chứa ma trận a , số thứ j trên dòng i là $a_{i,j}$

Dữ liệu đảm bảo $a_{i,i} = 0$, $a_{i,j} + a_{j,i} = 1 \forall i \neq j$

Kết quả

- Dòng đầu chứa độ dài chuỗi trận tìm được: k
- Dòng tiếp theo chứa chỉ số của k máy theo thứ tự trong chuỗi tìm được

Ví dụ

stdin	stdout
4	4
0 1 0 1	1 2 3 4
0 0 1 1	
1 0 0 1	
0 0 0 0	

Hạn chế

- Subtask 1: $1 \leq n \leq 20$
- Subtask 2: $21 \leq n \leq 1000$

Bài H. NIMGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Các học sinh trong lúc rảnh rỗi đã nghĩ ra một trò chơi như sau:

- Có n đồng sỏi, đồng thứ i có a_i viên, hai người chơi sẽ luân phiên nhau thực hiện lượt chơi
- Đến lượt mình, người chơi sẽ chọn một đồng sỏi và lấy đi một số sỏi tùy ý, ít nhất là một viên và nhiều nhất là toàn bộ số sỏi trong đồng đó
- Ai không thể thực hiện được lượt chơi của mình thì thua cuộc

Bạn được cho trước các thông tin của trò chơi và có quyền chọn lượt chơi, hãy giành chiến thắng trong trò chơi này!

Tương tác:

- Đầu tiên bạn phải đọc vào số n và dãy a_1, a_2, \dots, a_n từ đầu vào chuẩn
- Tiếp theo bạn in ra 0/1 tương ứng là muốn đi sau hay đi trước
- Ở lượt chơi của bạn, bạn cần in ra đầu ra chuẩn hai số i x ($1 \leq i \leq n, 1 \leq x \leq a_i$) thể hiện rằng bạn sẽ bốc x viên ở đồng thứ i
- Ở lượt chơi của máy, bạn phải đọc vào từ đầu vào chuẩn hai số i x ($1 \leq i \leq n, 1 \leq x \leq a_i$) thể hiện rằng máy sẽ bốc x viên ở đồng thứ i
- Hai đầu thủ sẽ luân phiên thực hiện lượt chơi của mình cho đến khi trò chơi kết thúc

Lưu ý bạn cần flush stdout (hoặc endl) sau mỗi dòng in ra để tương tác được với trình chấm. Khi một trong hai người chơi thắng cuộc, bạn cần ngắt chương trình của mình.

Ví dụ

<code>stdin</code>	<code>stdout</code>
3 1 2 3	

Hạn chế

- $n \leq 10^6$. $0 \leq a_i \leq 10^6$. Tổng các a_i không quá 10^6
- Có 50% số test với $n \leq 1000$. Tổng các a_i không quá 1000

02/08:

Bài I. LICS

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Hai đài quan sát thiên văn vừa thu nhận được tín hiệu từ một ngôi sao mới. Tín hiệu ở đài quan sát thứ nhất có thể mô tả bởi dãy số nguyên $a = a_1, a_2, \dots, a_n$. Tín hiệu ở đài quan sát thứ hai có thể mô tả bởi dãy số nguyên $b = b_1, b_2, \dots, b_m$. Các nhà khoa học tin rằng, tín hiệu từ một ngôi sao mới khi biểu diễn sẽ có dạng một dãy tăng dần, các số khác trong dãy có thể là do nhiễu trong quá trình quan sát. Để lọc nhiễu, với mỗi đài quan sát, người ta sẽ xoá đi một số phần tử trong dãy số tín hiệu (có thể không xoá phần tử nào) sao cho dãy còn lại là tăng dần. Công việc này sẽ được thực hiện ở cả hai đài quan sát, sau đó so sánh kết quả với nhau. Nếu hai dãy kết quả từ hai đài quan sát là giống nhau, dãy đó sẽ được gọi là một dãy tín hiệu đáng tin. Cụ thể hơn, dãy tín hiệu đáng tin là một dãy số, vừa là một dãy tăng vừa là một dãy con chung của hai dãy a và b .

Yêu cầu: Hãy tìm độ dài của dãy tín hiệu đáng tin dài nhất.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n, m ($n, m \leq 100$);
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^6$);
- Dòng thứ ba chứa m số nguyên b_1, b_2, \dots, b_m ($0 \leq b_i \leq 10^6$).

Kết quả

Ghi một số nguyên duy nhất là độ dài của dãy tín hiệu đáng tin dài nhất.

Ví dụ

stdin	stdout
3 5 1 2 2 2 2 5 3 5	1

Bài J. NUM19

Hạn chế thời gian: 2 giây

Cho hai số nguyên dương L, R . Hãy đếm xem có bao nhiêu số nguyên dương x thuộc đoạn $[L, R]$ hết cho 19; sao cho ở dạng biểu diễn thập phân, x không chứa hai chữ số nào có tổng chia hết cho 3.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($1 \leq T \leq 10^5$);
- Mỗi test được mô tả trên 2 dòng là L và R ($1 \leq L \leq R \leq 10^{10000}$).

Tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^6 .

Kết quả

Với mỗi testcase, ghi trên một dòng số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007.

Ví dụ

test	answer
2	4
1 100	2
101 200	

Giải thích

Các số thỏa mãn testcase 1 là: 19, 38, 76, 95.

Hạn chế

- Có 50% số test với $R \leq 10^6$;
- Có 30% số test với tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^3 ;
- Có 20% số test với ràng buộc gốc.

Bài K. NOTHING

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Bài này không có đề. Học sinh tự nghĩ ra đề để giải.

Dữ liệu vào

Chứa ba số nguyên dương n m k ($n, m, k \leq 1000$).

Kết quả

Ghi yes hoặc no là kết quả bài toán.

Ví dụ

<code>stdin</code>	<code>stdout</code>
1 1 1	yes

Hạn chế

- Subtask 1: Submit 300 lần
- Subtask 2: Submit 32 lần
- Subtask 3: Submit 17 lần

Bài L. MXTREE

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Cho một cây có gốc có n đỉnh. Mỗi đỉnh có một mã định danh id và một trọng số $w(id)$. Gốc của cây có định danh là r . Có Q thao tác dạng:

- 0 $p\ id\ w$: Thêm một đỉnh mới với định danh id , trọng số w và nhận p làm nút cha;
- 1 $id\ a$: Tìm $\min(a \wedge w)$ và $\max(a \wedge w)$ với w là trọng số của một đỉnh nào đó trên đường đi đơn từ id đến r .

Dữ liệu vào

- Dòng đầu ghi số đỉnh ban đầu và số thao tác: $n\ Q$;
- Dòng thứ hai mô tả đỉnh gốc: $r\ w(r)$;
- $n - 1$ dòng tiếp theo, mỗi dòng mô tả một đỉnh của cây: $id\ p\ w(id)$ là định danh, định danh của đỉnh cha, trọng số;
- Q dòng tiếp theo, mỗi dòng mô tả một thao tác, gồm 3 hoặc 4 số nguyên đã được mã hóa. Để giải mã, số s sẽ được thay bằng $s \wedge premin \wedge premax$. Ở đây $premin$, $premax$ là kết quả trước đó hoặc 0 0 nếu chưa có thao tác loại 1 nào.

Các định danh được đảm bảo khác nhau nhưng không nhất thiết tạo thành hoán vị của $[n]$. Dữ liệu đảm bảo hợp lệ.

Kết quả

Với mỗi thao tác loại 1, in ra hai số trên một dòng.

Ví dụ

stdin	stdout
7 3	2 5
1 3	0 7
2 1 1	
3 1 2	
4 2 3	
5 2 5	
6 3 4	
7 3 6	
1 6 1	
7 2 15 5	
6 15 5	

Hạn chế

- $1 \leq n, Q \leq 10^5$; $1 \leq id, w, a < 2^{31}$;
- 50% số test có $1 \leq n, Q \leq 5000$.

Bài M. GAME3579

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Học sinh ở trại hè rất hứng thú với các con số, và cả trò chơi nữa, vì vậy ban tổ chức đã nghĩ ra trò chơi sau:

- Có n viên bi
- Mỗi lượt chơi, người chơi lấy đi ba hoặc năm hoặc bảy hoặc chín viên bi (tất nhiên là nếu có đủ bi)
- Hai người luân phiên nhau thực hiện lượt chơi. Ai không thực hiện được lượt chơi nữa (tức số bi còn lại bé hơn ba) thì thua cuộc

Xác định ai là người chiến thắng, biết rằng cả hai đều chơi rất giỏi

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($1 \leq T \leq 10^5$)
- T dòng tiếp theo, mỗi dòng chứa một số: n ($1 \leq n \leq 10^{18}$)

Kết quả

Gồm T dòng, mỗi dòng ghi 1/2 tương ứng là người đi trước/sau thắng

Ví dụ

stdin	stdout
4	1
11	2
12	2
13	1
100	

Hạn chế

- Có 50% số test với $n \leq 10^5$

Bài N. KBIN

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 512MB

Cho hai số nguyên dương n, k . Xét các xâu nhị phân độ dài n mà không có hai bit 1 nào kề nhau, hãy tìm xâu thứ k theo thứ tự từ điển.

Dữ liệu vào

Gồm một dòng chứa hai số n, k ($1 \leq n \leq 43, 1 \leq k \leq 10^9$).

Kết quả

Nếu số lượng xâu thỏa mãn bé hơn k , in ra -1. Ngược lại, in ra một xâu nhị phân độ dài n không có hai bit 1 kề nhau, là xâu thứ k tính theo thứ tự từ điển.

Ví dụ

stdin	stdout
4 7	1001

Giải thích

Có 8 xâu nhị phân độ dài 4 mà không có hai bit 1 kề nhau, được liệt kê theo thứ tự từ điển là: 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010. Xâu thứ 7 trong dãy là 1001.

Hạn chế

- Có 20% test với $n \leq 20$;
- Có 30% test với $k \leq 10^5$;
- Có 50% test với ràng buộc gốc.

Bài O. ONETHING

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Nếu bài không đề khiến học sinh bối rối, thì đây là một bài có đề hẫng hoi. Đề bài như sau:

Ở mỗi test, bạn sẽ được đọc vào ba số nguyên dương n, m, k . Hãy in ra giá trị của $n + m + k$ nhưng ... của test trước đó, hoặc in ra -1 nếu đây là test đầu tiên.

Dữ liệu vào

Chứa ba số nguyên dương $n\ m\ k$ ($n, m, k \leq 1000$).

Kết quả

Ghi $n + m + k$ của test trước đó, hoặc -1 nếu đây là test đầu tiên.

Ví dụ

stdin	stdout
1 1 1	-1
1 1 2	3

Hạn chế

- Bộ test không có hai input nào giống nhau.
- Subtask 1: Submit 300 lần
- Subtask 2: Submit 32 lần
- Subtask 3: Submit 17 lần

Bài P. XQUERY

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 2 giây

Cho S là một tập các số nguyên không âm (các số có thể xuất hiện nhiều lần). Ban đầu tập S rỗng, cần thực hiện Q truy vấn trên S , mỗi truy vấn thuộc một trong các dạng sau:

- 0 x : Thêm một số x vào S ($0 \leq x \leq 10^5$). Nếu x đã có trong S thì truy vấn này vẫn được thực hiện như thường.
- 1 x : Xóa một số x khỏi S ($0 \leq x \leq 10^5$). Nếu $x \notin S$ thì truy vấn này không cần làm gì. Nếu x xuất hiện nhiều lần trong S thì truy vấn này chỉ xóa đi 1.
- 2 a : Thay thế tất cả phần tử của S , x thay bằng $x \wedge a$ với \wedge là phép toán xor/nim/hoặc triệt tiêu ($0 \leq a \leq 10^5$)
- 3 k : Tính tổng của k phần tử nhỏ nhất trong S ($0 \leq k \leq |S|$)

Dữ liệu vào

- Dòng đầu tiên chứa Q ($1 \leq Q \leq 10^5$)
- Q dòng tiếp theo mỗi dòng chứa một truy vấn: $t \ val$

Kết quả

Với mỗi truy vấn loại 3, hãy in ra tổng của k phần tử nhỏ nhất trong S trên một dòng

Ví dụ

stdin	stdout
6	6
0 1	1
2 2	
0 3	
3 2	
2 2	
3 1	

04/08:

Bài Q. FSUM

File dữ liệu vào: FSUM.inp
File kết quả: FSUM.out
Hạn chế thời gian: 1 giây

Tính $\lfloor \frac{n}{1} \rfloor + \lfloor \frac{n}{2} \rfloor + \dots + \lfloor \frac{n}{m} \rfloor$

Dữ liệu vào

- Gồm một số dòng, mỗi dòng chứa 2 số nguyên dương n m
- Kết thúc file là 2 số 0, không cần phải xử lý trường hợp này

Kết quả

- Mỗi dòng ghi kết quả cho một trường hợp

Ví dụ

FSUM.inp	FSUM.out
5 5	10
6 2	9
10 20	27
1000 900	6969
1000000000000 1000000000000	27785452449086
0 0	

Hạn chế

- $1 \leq n \leq 10^{12}$

Bài R. WARRAY

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Trọng số của một dãy số nguyên là tổng của dãy nhân với số phần tử của dãy. Ví dụ trọng số của dãy $(4, -2, 3, 1)$ là $6 \times 4 = 24$.

Cho dãy số nguyên $a = a_1, a_2, \dots, a_n$. Hãy tách a thành các dãy con (không nhất thiết kề nhau) sao cho tổng trọng số của các dãy con là lớn nhất có thể.

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n ;
- Dòng thứ hai chứa n số nguyên không âm: a_1, a_2, \dots, a_n ($-10^8 \leq a_i \leq 10^8$);

Kết quả

Ghi một số nguyên là tổng trọng số lớn nhất tìm được.

Ví dụ

<code>stdin</code>	<code>stdout</code>
6 3 -1 -5 3 -2 6	40

Hạn chế

- Có 12% số test với $n \leq 20$;
- Có 28% số test với $n \leq 1000$;
- Có 60% số test với $n \leq 10^5$;

Bài S. SOMETHING

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Hùng đang cố gắng làm bài onething thì nhận ra, anh buộc phải tìm ra input của tất cả các test. Anh đã sửa code của mình và submit nhiều lần, sau đó dựa vào kết quả chấm để dò ra input. Hệ thống chấm có thể phản hồi cho từng test với các khả năng: WA, MLE, RE, TLE (có thể có cả OK nhưng vì không phân biệt được nên Hùng không sử dụng khả năng này). Hùng code như sau:

- Đọc vào n, m, k ;
- Tính ra $x = f$ với f là một công thức được code trong mã nguồn, biểu thức kiểu số nguyên chỉ chứa các ký tự: $+ - * / \% () 0..9nmk$
- Nếu $x \leq 0$, cậu sẽ in ra "iloveyou" (tức là cố tình để WA);
- Nếu $x = 1$, cậu sẽ cấp phát bộ nhớ liên tục (tức là cố tình để MLE);
- Nếu $x = 2$, cậu sẽ thực hiện phép chia cho 0 (tức là cố tình để RE);
- Nếu $x > 2$, cậu sẽ chạy một vòng lặp vô hạn (tức là cố tình để TLE);

Với mỗi lần sửa, cậu sẽ sửa lại công thức f trong mã nguồn sau đó submit, sau đó nhận kết quả chấm.

Yêu cầu: Hãy giúp Hùng tự động hoá công việc này.

Tương tác:

- Để sửa mã nguồn và submit lại, bạn cần in ra theo định dạng: edit f với ý nghĩa: Sửa biểu thức trong nguồn thành f sau đó submit. Sau đó, bạn cần đọc vào kết quả chấm từ đầu vào chuẩn theo định dạng: Một xâu có độ dài bằng số lượng test, chỉ chứa các ký tự W, M, R, T với ý nghĩa: Các test tương ứng có kết quả chấm là WA, MLE, RE, TLE. Xâu f chỉ chứa các ký tự $+ - * / \% () 0..9nmk$ và được viết liền nhau.
- Nếu đã dò ra toàn bộ input của tất cả các test, bạn cần in ra theo định dạng: answer $n_1 m_1 k_1 n_2 m_2 k_2 \dots n_t m_t k_t$ với ý nghĩa: t là số test, test thứ i có input là n_i, m_i, k_i .

Hệ thống đảm bảo phản hồi nhất quán và chính xác. Số điểm của bạn phụ thuộc vào số lần edit.

Ví dụ

Lưu ý, VD chỉ mang tính minh hoạ cho cách thức tương tác.

stdin	stdout
edit (n*2+m/2-k*k)%4	TWTR
edit (n*2+m/2-k*k)%2	WMMW
edit n*(3+m)/2-k/4%10	WWWW
answer	
1 1 1	
1 2 3	
3 2 1	
3 1 2	

Hạn chế

- Trong tất cả các bộ test: số test (độ dài xâu) không quá 100; $1 \leq n, m, k \leq 1000$
- Subtask 1: Bộ test chỉ có 1 test;
- Subtask 2: Bộ test thoả mãn $m = k = 0$;
- Subtask 3: Ràng buộc gốc.

Điểm của bạn sẽ là $\min(1, \frac{16}{x+1})$ với x là số lần edit.

Bài T. APL

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Một chiếc máy bay đang ở độ cao h_1 . Phi công cần điều khiển để máy bay đạt độ cao h_2 trong đúng n giây. Tại mỗi giây, phi công có thể điều khiển bởi một trong ba lệnh: Tăng độ cao lên 1, giảm độ cao đi 1, giữ nguyên độ cao. Hãy đếm số cách điều khiển khác nhau. Biết rằng máy bay có thể chạm vào mặt đất (độ cao 0) nhưng không thể đạt độ cao âm. Hai cách điều khiển được cho là khác nhau nếu tồn tại i , $1 \leq i \leq n$ sao cho lệnh điều khiển ở thời điểm thứ i trong hai cách trên là khác nhau.

Dữ liệu vào

- Gồm ba số tự nhiên: h_1 h_2 n

Kết quả

- In ra phần dư của số cách điều khiển khi chia cho $10^9 + 7$

Ví dụ

<code>stdin</code>	<code>stdout</code>
0 0 6	51

Hạn chế

- $n, h_1, h_2 \leq 10^5$
- Subtask 1: $h_1, h_2 \geq n$
- Subtask 2: $h_1 = h_2 = 0$
- Subtask 3: Ràng buộc gốc

Bài U. TANK

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Xe tăng là một phương tiện có cách di chuyển rất đặc biệt. Các bánh xe của nó trải dài trên nền đất để tăng diện tích tiếp xúc, từ đó giảm áp lực lên nền. Giả sử xe tăng đang muốn đi từ A đến B, ta có thể chia đoạn đất này thành n đoạn nhỏ, đoạn thứ i có độ cứng a_i . Một xe tăng chiều dài L , khối lượng M có thể đi qua nếu tại mọi thời điểm, nó luôn đứng trên vùng đất có tổng độ cứng lớn hơn M (có nghĩa là mọi đoạn con liên tiếp độ dài L của dãy a đều phải có tổng lớn hơn hoặc bằng M). Cho biết khối lượng M của xe tăng, hãy tính chiều dài L nhỏ nhất có thể có của nó để xe tăng đi qua được vùng đất này.

Dữ liệu vào

- Dòng đầu chứa: M n
- Dòng tiếp theo chứa: a_1 a_2 ... a_n

Dữ liệu đảm bảo tổng của mảng a lớn hơn hoặc bằng M

Kết quả

Một số nguyên duy nhất là chiều dài ngắn nhất có thể của xe tăng

Ví dụ

stdin	stdout
6 5 3 2 1 4 5	3

Hạn chế

- $1 \leq n \leq 10^5$, $1 \leq a_i, M \leq 10^9$
- $n \leq 1000$ với ít nhất 50% test

Bài V. TWOPATH

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho một cây với các cạnh có trọng số là các số nguyên không âm. Trọng số của một đường đi đơn là xor của các trọng số các cạnh trên đường đi đó. Hãy tìm hai đường đi đơn có trọng số bằng nhau (Hai đường đi phải khác nhau, và mỗi đường đi phải chứa ít nhất một cạnh).

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n là số đỉnh của cây ($3 \leq n \leq 3 \times 10^5$);
- Mỗi dòng trong số $n - 1$ dòng tiếp theo chứa ba số nguyên u, v, w cho biết có một cạnh nối u và v với trọng số w ($0 \leq w \leq 2^{20}$).

Các đỉnh của cây được đánh số từ 1 đến n .

Kết quả

Ghi bốn số nguyên dương u, v, x, y cho biết đường đi đơn từ u đến v và đường đi đơn từ x đến y có trọng số bằng nhau. Hai đường đi này phải khác nhau, tức là tồn tại một đỉnh nằm trên đường đi này nhưng không nằm trên đường đi kia. Nếu có nhiều kết quả, chỉ cần in ra một trong số đó. Nếu không có kết quả, in ra -1.

Ví dụ

stdin	stdout
6 1 2 2 1 3 3 2 4 4 2 5 1 3 6 7	1 3 1 5

Hạn chế

- Có 10% số test với $n \leq 100$;
- Có 15% số test với $n \leq 500$;
- Có 20% số test với $n \leq 5000$;
- Có 25% số test với cây có dạng một đường thẳng;
- Có 30% số test còn lại không có ràng buộc gì thêm.

Bài W. CARDGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho một bộ bài 52 lá thông thường. Các lá bài được xác định bởi số và chất. Số được đánh là 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A. Chất được đánh là P, C, T, H. Ví dụ quân Át bích là AP, quân 7 tép là 7C. An và Bình biểu diễn một trò ảo thuật như sau:

- An lấy ra một bộ bài, bảo khán giả chọn 6 lá tùy ý rồi đưa lại cho An (không cho Bình biết)
- An đưa cho Bình 5 lá và giữ lại một lá (không cho Bình biết lá này)
- Bình đoán ra lá bài An đang giấu trước sự trầm trồ của khán giả

Dĩ nhiên họ là một cặp diễn ăn ý, trước đó họ đã thống nhất chiến thuật với nhau. Nhiệm vụ của bạn là giả lập lại trò ảo thuật này, bằng cách viết hai class (hoặc struct) **An** và **Bình**:

class **An** gồm:

- Hàm khởi tạo mặc định.
- Hàm `vector<string> pick(vector<string> X)` để chọn các lá bài để đưa cho Bình: Nhận đầu vào *X* là 6 lá bài mà khán giả chọn, trả ra *Y* là danh sách các lá bài sẽ đưa cho Bình.

class **Bình** gồm:

- Hàm khởi tạo mặc định.
- Hàm `string guess(vector<string> Y)` để đoán lá bài còn lại mà An đang giữ: Nhận đầu vào *Y* là kết quả trả về của hàm `pick` nói trên, trả ra một lá bài là lá mà An đang giữ.

Các hàm trên cần để **public**. Học sinh được phép viết thêm các hàm và biến khác nếu cần, tùy thuộc vào thiết kế của mình. Khi chấm bài, trình chấm sẽ giả lập buổi biểu diễn và sử dụng các hàm do học sinh viết. Việc thực hiện trò ảo thuật này như thế nào là do học sinh tự thiết kế, trình chấm chỉ bảo đảm sẽ đưa chính xác **vector Y** trả về của hàm `pick` cho hàm `guess`, ngoài ra sẽ không can thiệp gì vào mô hình của học sinh.

Lưu ý, yêu cầu của bài là thiết kế lớp nên học sinh không cần viết hàm **main**. Thay vào đó, cần thêm dòng `#include "main.h"` vào cuối bài làm để chấm điểm. Xem mã nguồn mẫu dưới đây:

```
#include "bits/stdc++.h"
using namespace std;

class An {
public:
    An() {}

    std::vector<std::string> pick(std::vector<std::string> a) {
        //your code here
    }
};

class Bình {
public:
    Bình() {}

    std::string guess(std::vector<std::string> b) {
        //your code here
    }
};

#include "main.h"
```


05/08:

Bài Y. BR1

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng
- Nếu A là biểu thức ngoặc đúng thì (A) cũng là dãy ngoặc đúng
- Nếu A và B là biểu các thức ngoặc đúng thì AB cũng là biểu thức ngoặc đúng

Tiến hành sắp xếp tất cả các dãy ngoặc đúng độ dài n theo thứ tự từ điển (với $'(' < ')'$), sau đó đánh số thứ tự cho các xâu bắt đầu từ 1 Cho S là một biểu thức ngoặc đúng có độ dài n . Tìm số được đánh cho S

Dữ liệu vào

- Dòng duy nhất chứa xâu S

Kết quả

- Chứa số nguyên dương là thứ tự của S , chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
<code>()()()</code>	5

Hạn chế

- $1 \leq |S| \leq 10^3$, $|S|$ chẵn
- Subtask 1: $n \leq 10$
- Subtask 2: Ràng buộc gốc

Bài Z. BR2

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng
- Nếu A là biểu thức ngoặc đúng thì (A) cũng là dãy ngoặc đúng
- Nếu A và B là biểu các thức ngoặc đúng thì AB cũng là biểu thức ngoặc đúng

Cho S là một biểu thức ngoặc đúng có độ dài n . Tiến hành sắp xếp tất cả các dãy ngoặc đúng độ dài n theo thứ tự từ điển (với ' $' < ')' > '('$ '), sau đó đánh số thứ tự cho các xâu bắt đầu từ 1. Tìm số được đánh cho S . Bạn cần trả lời nhiều truy vấn như vậy, độ dài xâu S ở các truy vấn có thể khác nhau

Dữ liệu vào

- Dòng đầu tiên chứa Q là số lượng truy vấn
- Q dòng tiếp theo mỗi dòng chứa một xâu S

Kết quả

- Gồm Q dòng. Mỗi dòng chứa số nguyên dương là thứ tự của S với test tương ứng, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
3	5
()()()	1
()	2
()()	

Hạn chế

- $1 \leq |S|, Q \leq 10^3$, $|S|$ chẵn
- Subtask 1: Tất cả các xâu S có cùng độ dài
- Subtask 2: Ràng buộc gốc

Bài A. HRTRANS

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Quân đội của hành tinh XYZ đang chiến đấu với một hành tinh khác. Họ có một đơn vị chỉ huy và nhiều đơn vị chiến đấu được bố trí ở các vị trí chiến lược. Trước khi bắt đầu chiến dịch tấn công, đơn vị chỉ huy sẽ chọn thời điểm tấn công cho từng đơn vị chiến đấu, sau đó phát lệnh tới từng đơn vị chiến đấu (mỗi đơn vị có thể có một thời điểm khác nhau, để tiếp ứng cho nhau và đánh chiến thuật với địch).

Bờm cần thiết kết hệ thống truyền tin cho quân đội này. Hệ thống gồm có hai loại mô-đun, mô-đun loại A đặt ở đơn vị chỉ huy và các mô-đun loại B đặt ở các đơn vị chiến đấu ngoài mặt trận. Khi đã lên kế hoạch tác chiến xong, A sẽ cần mã hoá thời điểm tấn công của từng đơn vị chiến đấu để gửi tới các B. Việc truyền tin sử dụng sóng vô tuyến vốn đã quen thuộc, tuy nhiên có thể do bị nhiễu hoặc do bị địch quấy phá nên tín hiệu truyền đi có thể bị sai không quá một bit. Việc còn lại chỉ là mã hoá thông tin thành dạng nhị phân để gửi đi và giải mã nó khi nhận được, sao cho dù có bị sai một bit hay không thì vẫn bảo đảm giải mã chính xác. Bờm nhờ bạn xây dựng hai lớp `Captain` và `Soldier` để mã hoá và giải mã thông tin thời điểm trong ngày.

Lớp `Captain`:

- Khởi tạo mặc định.
- Hàm `std::string encode(std::string time)`: nhận vào một thời điểm với định dạng `hh:mm:ss`, và trả về một chuỗi nhị phân mã hóa thời điểm đó. Độ dài chuỗi mã hóa càng ngắn thì học sinh sẽ càng được điểm cao.

Lớp `Soldier`:

- Khởi tạo với mã số đơn vị: `Soldier(int id)`
- Hàm `std::string decode(std::string s)`: nhận vào một chuỗi nhị phân `s` và trả về thời điểm tương ứng với định dạng `hh:mm:ss`.

Các hàm trên cần để `public`. Học sinh được phép viết thêm các hàm và biến khác nếu cần, tùy thuộc vào thiết kế của mình. Khi chấm bài, trình chấm sẽ giả lập môi trường của hệ thống truyền tin của Bờm và sử dụng các hàm mã hoá, giải mã do học sinh viết. Điểm của học sinh cho một test là $\min(1, \frac{23}{L})$ với L là độ dài xâu mã hoá của học sinh.

Lưu ý, yêu cầu của bài là thiết kế lớp nên học sinh không cần viết hàm `main`. Thay vào đó, cần thêm dòng `#include "main.h"` vào cuối bài làm để chấm điểm. Xem mã nguồn mẫu dưới đây:

```
#include "bits/stdc++.h"

class Captain {
public:
    Captain() {}

    std::string encode(std::string time) {
    }
};

class Soldier {
private:
    int id;

public:
    Soldier(int id) : id(id) {}

    std::string decode(std::string s) {
    }
};

#include "main.h"
```

Bài B. PINZ

File dữ liệu vào:	<code>standard input</code>
File kết quả:	<code>standard output</code>
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Chung cư Vui Vẻ vừa lắp đặt các tấm pin mặt trời lên mái của tòa nhà. Mái có thể được mô tả như một hình chữ nhật có chiều dài n đơn vị và chiều rộng 1 đơn vị. Hai cạnh chiều dài của mái được ký hiệu là U và D . Có k tấm pin đã được lắp đặt. Các tấm pin đều có hình thang với hai đáy trùng lên cạnh chiều dài của mái. Tấm pin thứ i có đáy trên là đoạn thẳng nối từ vị trí a_i đến b_i của cạnh U ($0 \leq a_i \leq b_i \leq n$), và đáy dưới là đoạn thẳng nối từ vị trí c_i đến d_i của cạnh D ($0 \leq c_i \leq d_i \leq n$).

Sau khi lắp đặt xong, người ta cần đấu nối dây dẫn để thu gom điện từ các tấm pin này. Hai tấm pin có điểm chung (điểm chung có thể nằm trên biên hoặc bên trong hình thang) có thể được liên kết để dùng chung. Việc liên kết phải thỏa mãn nếu tấm pin thứ i liên kết với tấm pin thứ j và tấm pin thứ j liên kết với tấm pin thứ k thì tấm pin thứ i liên kết với tấm pin thứ k , điều này là để đảm bảo cung cấp điện đều đặn khi có sự cố. Lúc này, mỗi một nhóm các tấm pin được liên kết chỉ cần đấu nối vào một điểm để thu gom điện từ tất cả các tấm pin này. Hãy giúp ban quản lý liên kết các tấm pin để cực tiểu số điểm cần thu gom. Cụ thể hơn, cần chia các tấm pin thành ít nhất các nhóm sao cho trong mỗi nhóm, các tấm pin đôi một có điểm chung.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n, k ;
- Dòng thứ i trong số k dòng tiếp theo chứa a_i, b_i, c_i, d_i .

Kết quả

Ghi một số nguyên dương duy nhất là số nhóm ít nhất.

Ví dụ

standard input	standard output
10 4 0 4 0 4 1 7 1 7 3 10 3 10 5 9 5 9	2
11 4 2 3 0 5 5 8 2 6 7 9 7 9 8 11 9 10	2

Hạn chế

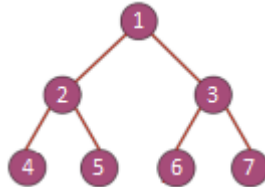
- Trong tất cả các test: $n, k \leq 10^5$;
- Có 16% test với $k \leq 10^3$;
- Có 32% test với $a_i = c_i$ và $b_i = d_i$;

- Có 32% test với $a_i = b_i$ và $c_i = d_i$;
- Có 20% test tiếp theo với ràng buộc gốc.

Bài C. WTREE

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cây là một đơn đồ thị vô hướng liên thông không có chu trình. Giữa hai đỉnh x, y trên cây, luôn luôn tồn tại và duy nhất một đường đi đơn giữa chúng



Hệ thống mạng trên đất nước xyz có dạng hình cây, mỗi nút mạng là một đỉnh và mỗi cặp nối là một cạnh. Các đỉnh được đánh số từ 1 đến n , các cạnh được đánh số từ 1 đến $n - 1$. Ban đầu mọi cặp nút mạng x, y đều liên lạc được với nhau. Những kẻ tấn công mạng (hacker) ở đây thường thực hiện các vụ tấn công... vật lý. Theo đó, chúng sẽ cắt đứt một số cặp nối theo một thứ tự định trước. Khi một cặp nối bị cắt đi, sẽ có thêm một số cặp nút mạng không liên lạc được với nhau nữa, số cặp như vậy gọi là hiệu quả của lát cắt. Hãy giúp nhóm hacker tính toán hiệu quả của từng lát cắt

Dữ liệu vào

- Dòng đầu ghi số nút mạng: n
- $n - 1$ dòng tiếp theo, mỗi dòng ghi một cặp nối: u, v
- Dòng tiếp theo ghi số cặp nối sẽ bị cắt: Q
- Q dòng tiếp theo, mỗi dòng ghi chỉ số một cặp nối bị cắt: i

Dữ liệu đảm bảo các cặp nối bị cắt chưa từng bị cắt trước đó và được liệt kê theo thứ tự sẽ cắt

Kết quả

- Gồm Q dòng ghi hiệu quả của Q lát cắt theo thứ tự đầu vào

Ví dụ

stdin	stdout
7	6
1 2	8
1 3	3
2 4	2
2 5	
3 6	
3 7	
4	
3	
1	
5	
2	

Hạn chế

- Subtask 1: $1 \leq n, Q \leq 1000$
- Subtask 2: $1 \leq n, Q \leq 10^5$

Bài D. BINACOUNT

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho một chuỗi S chỉ chứa các ký tự a, b . Có thể biến đổi chuỗi S bằng các thao tác:

- Thay một ký tự a thành 10
- Thay một ký tự b thành 01
- Thay hai ký tự liên tiếp ab thành 101
- Thay hai ký tự liên tiếp ba thành 010

Các thao tác có thể được sử dụng nhiều lần với thứ tự tùy ý và vị trí tùy ý. Hãy đếm số chuỗi nhị phân (chuỗi chỉ chứa các ký tự 0 hoặc 1) khác nhau có thể tạo ra.

Dữ liệu vào

Gồm một dòng duy nhất chứa chuỗi S chỉ gồm các ký tự a và b .

Kết quả

Ghi một số nguyên duy nhất là số lượng chuỗi nhị phân có thể tạo ra từ S bằng các thao tác trên. Do kết quả có thể rất lớn, chỉ cần in ra phần dư khi chia cho 1000000007 .

Ví dụ

<code>stdin</code>	<code>stdout</code>
<code>aabbabab</code>	<code>16</code>

- Trong tất cả các test: $1 \leq |S| \leq 10^5$.
- Có 20% số test với $|S| \leq 10$;
- Có 20% số test với $|S| \leq 20$;
- Có 20% số test với $|S| \leq 1000$;
- Có 40% số test với ràng buộc gốc.

Bài E. PLUS

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho hai số nguyên A và B . Cần biến đổi để hai số này bằng nhau bằng cách thực hiện không quá 5000 lệnh, mỗi lệnh có dạng " $A+ = B$ " hoặc " $B+ = A$ " hoặc " $A+ = A$ " hoặc " $B+ = B$ " với ý nghĩa là "tăng A lên một lượng bằng B " hoặc "tăng B lên một lượng bằng A " hoặc "tăng A lên một lượng bằng A " hoặc "tăng B lên một lượng bằng B ".

Dữ liệu vào

- Gồm hai số nguyên A B

Kết quả

- Dòng đầu ghi số lượng biến đổi Q
- Theo sau là Q dòng, mỗi dòng ghi một biến đổi

Ví dụ

stdin	stdout
2 3	4 B+=B B+=A A+=A A+=A

Hạn chế

- $1 \leq A, B \leq 10^{18}$

Bài F. COCKROACH

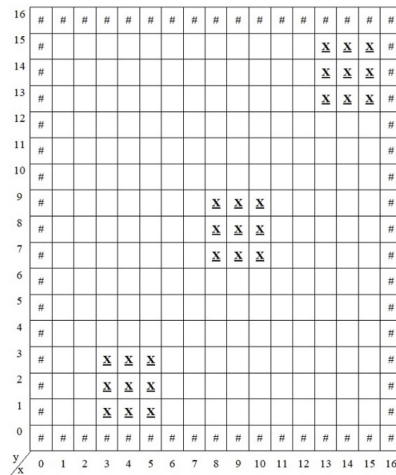
File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Có N con gián sống trong một cái tủ, được đánh số từ 0 đến $N - 1$. Tủ có thể mô tả như một ma trận $K \times K$ với các ô vuông đơn vị. Tổ của con gián thứ i là hình vuông có góc trái dưới là ô $(x_0 + i \times v_x, y_0 + i \times v_y)$ và cạnh dài M đơn vị. Dữ liệu đảm bảo $M < v_x, v_y \leq 2M$ và tổ của các con gián đều nằm gọn trong tủ. Tiếc là tủ đóng và bạn chỉ biết kích thước của nó. Bạn cần tìm ra các tổ này, tức là tìm N, M, x_0, y_0, v_x, v_y . Bạn có một máy dò có thể cho biết trong hai điểm A và B, điểm nào gần một tổ gián hơn, và được dò không quá 160 lần. Gần hơn ở đây được hiểu là khoảng cách Mahatan nhỏ nhất đến một trong số các ô thuộc một tổ gián nào đó là nhỏ hơn.

Tương tác:

- Đầu tiên bạn đọc vào K từ đầu vào chuẩn ($1 \leq K \leq 10^9$);
- Để dò, bạn in ra theo định dạng: $? x_A y_A x_B y_B$ ($0 \leq x_A, y_A, x_B, y_B \leq 10^9$). Sau đó bạn đọc vào một số nguyên từ đầu vào chuẩn, là -1/0/1 tương ứng A gần một tổ gián hơn, gần như nhau, B gần một tổ gián hơn.
- Để trả lời, bạn in ra theo định dạng: $! N M x_0 y_0 v_x v_y$ và kết thúc chương trình. Trong trường hợp $N = 1$, quy ước rằng $v_x = v_y = 0$.

Lưu ý: Cần flush(stdout) (hoặc cout « endl) sau mỗi dòng in ra để tương tác được với trình chấm.



Ví dụ

stdin	stdout
15	? 0 0 0 1
1	? 0 1 1 0
0	? 3 2 2 1
-1	? 3 1 4 2
0	? 12 11 12 11
0	! 3 3 3 1 5 6