Bài A. SFROG

File dữ liệu vào:	NULL
File kết quả:	NULL
Hạn chế thời gian:	1 giây

Nhiệm vụ của bạn trong bài này là bắn chết một con ếch. Để dễ hình dung, con ếch đang ở tọa độ x_f nào đó trên trục số nguyên. Ban đầu $-10^9 \leq x_f \leq 10^9$. Có hai thao tác có thể được thực hiện là

- Bắn vào tọa độ x
- Quan sát phạm vi $[L, H]$ để biết con ếch có ở trong phạm vi đó hay không

Mỗi lần bạn quan sát thấy con ếch, nó cũng sẽ thấy bạn và sẽ nhảy sang một vị trí khác, kề với vị trí đang đứng, tức là $x_f + 1$ hoặc $x_f - 1$. Việc nhảy sẽ diễn ra ngay sau khi bạn nhìn thấy nó

Hệ thống cung cấp thư viện "SFROG.h" để tương tác với máy, có sẵn các hàm sau:

- int query(int L,int H) trả về 1 hoặc 0 tương ứng thấy hoặc không thấy ếch khi quan sát đoạn [L,H] ($(-2\times10^9 \leq L \leq H \leq 2 \times 10^9)$). Nếu ếch đang đứng trong đoạn $[L, H]$, kết quả sẽ trả về 1, và ếch sẽ nhảy đến vị trí kề với vị trí nó đang đứng. Ngược lại, kết quả trả ra 0 và ếch vẫn đứng yên
- void shoot(int x) bắn vào x $(-2 \times 10^9 \leq x \leq 2 \times 10^9)$

Khi hàm shoot(x) được gọi (duy nhất 1 lần), chương trình sẽ tự ngắt và chấm điểm. Bạn sẽ nhận được điểm khi bắn trúng vị trí mà ếch đang đứng. Chương trình bị chấm là "Sai kết quả" nếu:

- Không gọi hàm shoot(x) hoặc gọi nhưng bắn không trúng
- Gọi đến hàm query(L,H) quá 40 lần
- Gọi đến một trong hai hàm nhưng tham số truyền vào không thỏa mãn điều kiện được kiểm tra

Để sử dụng được thư viện, phải có dòng khai báo #include"SFROG.h" ở đầu chương trình.

Bài B. RGAME

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 viên bi trên bàn, hai người chơi sẽ luân phiên nhau lấy
- Đến lượt mình, người chơi lấy đi một số viên bi từ số bi trên bàn, ít nhất là một viên và nhiều nhất là toàn bộ
- Lượt chơi đầu tiên phải lấy ít hơn n viên. Các lượt sau đó phải lấy không quá số bi vừa được lấy bởi đối thủ
- Ai không thể thực hiện được lượt chơi của mình thì thua cuộc

Bạn được cho trước n và quyền chọn lượt chơi, hãy giành chiến thắng trong trò chơi này! Bạn được cung cấp thư viện "RGAMELIB.h" để tương tác với máy, có sẵn các hàm sau:

- int get_n() trả về giá trị n
- void play(int x) thực hiện lấy đi x viên trên bàn ở lượt chơi của bạn
- int get_x() trả về số bi mà máy lấy ở lượt chơi của máy

Bạn sẽ bị xử thua nếu:

- Gọi đến hàm get_n() nhiều hơn 1 lần
- Gọi đến hàm play(x) hoặc get_x() trước khi gọi hàm get_n()
- Gọi hàm play(x) hai lần liên tiếp mà không có get_x() xen giữa
- Gọi hàm get_x() hai lần liên tiếp mà không có play(x) xen giữa
- Gọi hàm play(x) với tham số x không thỏa mãn luật chơi

Để sử dụng được thư viện, phải có dòng khai báo #include"RGAMELIB.h" ở đầu chương trình. Nếu trong quá trình chơi, một trong hai đấu thủ thua cuộc thì máy chấm sẽ ngắt chương trình và chấm điểm

Hạn chế

- $n \leq 10^6$
- Có 50% số test với $n \leq 1000$

Bài C. CARD

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

An và Bình diễn một trò ảo thuật như sau:

- An lấy ra một bộ bài (bộ bài 52 lá thông thường), bảo khán giả chọn 6 lá tùy ý và không cho Bình biết
- An đưa cho Bình 5 lá, giữ lại một lá và 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

Bài này không có Input hay Output thay vào đó bạn phải hoàn thành 2 hàm:

- vector<string> An(vector<string> a) mô tả chiến thuật của An: Nhận vào 6 lá bài mà khán giả chọn, trả ra 5 lá bài mà anh muốn đưa cho An, (dưới dạng một vector<string>).
- string Binh(vector<string> b) mô tả chiến thuật của Bình: Nhận vào các lá bài mà An đưa cho, trả ra một lá bài là dự đoán của anh về lá bài An đang giấu.

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.

Lưu ý là chương trình bạn nộp không cần có hàm main, cũng không cần đọc xuất dữ liệu. Trình chấm sẽ gọi riêng từng hàm và truyền tham số vào để chấm bài. Thứ tự hoạt động chấm như sau:

- Trình chấm đóng vai khán giả, chọn ngẫu nhiên 6 lá bài.
- Trình chấm gọi hàm An(.) và truyền 6 lá bài này vào, kết quả trả ra là một vector<string>.
- vector này được kiểm tra tính đúng đắn: Chứa 5 quân bài trong số 6 quân đầu vào.
- Sau đó vector được truyền vào cho hàm Binh(.), kết quả trả ra sẽ được kiểm tra.
- Trình chấm đảm bảo sẽ không sửa dữ liệu trong vector của bạn.

File mã nguồn nộp có thể chứa thêm các hàm, thủ tục, lớp, cấu trúc khác nếu cần dùng, nhưng bắt buộc phải chứa hai hàm yêu cầu và không có biến toàn cục nào được khai báo. Cần khai báo #include "CARD.h" ở đầu chương trình, thư viện này chứa các hàm của trình chấm và có sẵn các thư viện khác trong <bits/stdc++.h>. Xem file CARD_example.cpp (đính kèm trong mục đề bài trên hệ thống) để hiểu rõ hơn (lưu ý là trong file CARD_example, An và Bình chỉ chơi bừa, không phải chiến thuật đúng)

Ví dụ

- Khán giả chọn KC AC 10H 5T AH 8P
- An đưa cho Bình AH AC KC 5T 10H
- Bình đoán 8P

Bài D. 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 đống sỏi, đống 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 đống sỏi và lấy đi một số sỏi tùy ý, ít nhất là một viên và nhiều nhất là toàn bộ số sỏi trong đống 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! Bạn được cung cấp thư viện "NIMGAMELIB.h" để tương tác với máy, có sẵn các hàm sau:

- int get_n() trả về giá trị n
- int get_a(int i) trả về giá trị a_i
- void play(int i,int x) thực hiện lấy đi x viên từ đống thứ i ở lượt chơi của bạn
- void get_play(int& i, int& x) gán i và x lần lượt là chỉ số đống và số sỏi mà máy lấy ở lượt chơi của máy

Bạn sẽ bị xử thua nếu:

- Gọi hàm play(i,x) hai lần liên tiếp mà không có get_play(i,x) xen giữa
- Gọi hàm get_play(i,x) hai lần liên tiếp mà không có play(i,x) xen giữa
- Gọi hàm play(i,x) với tham số không thỏa mãn luật chơi

Để sử dụng được thư viện, phải có dòng khai báo #include"NIMGAMELIB.h" ở đầu chương trình. Xem file NIMGAME_example.cpp để hiểu rõ hơn cách tương tác. Nếu trong quá trình chơi, một trong hai đấu thủ thua cuộc thì máy chấm sẽ ngắt chương trình và chấm điểm

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 10^3