Bài A. 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 $a = a_1, a_2, \ldots, a_n$. Tín hiệu ở đài quan sát thứ hai có thể mô tả bởi dãy số nguyên $b = b_1, b_2, \ldots, b_m$. 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 $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^6$);
- Dòng thứ ba chứa m số nguyên $b_1, b_2, \ldots, b_m$ ($0 \leq b_i \leq 10^6$).

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

Hạn chế

- Subtask 1: $n \leq 20$.
- Subtask 2: $n \leq 50$.
- Subtask 3: $n \leq 100$.
- Subtask 4: $n \leq 500$.

Bài B. NETCOUNT

Hệ thống mạng trên hành tinh XYZ gồm có n nút mạng và m dây cáp; sao cho không có hai nút nào có nhiều hơn một dây cáp nối chúng, không có dây cáp nào nối một nút với chính nó, và tất cả các nút đều có thể truyền tin cho nhau. Thời gian truyền tin giữa hai nút là số lượng dây cáp trên đường đi ít dây cáp nhất giữa hai nút đó. Tiếc thay Bob đã đánh mất sơ đồ về hệ thống mạng, tuy nhiên anh có thể đo được thời gian truyền tin từ máy tính của anh (nút mạng số 1) đến tất cả các nút mạng khác. Hệ thống mạng thoả mãn: Đường đi ngắn nhất giữa 1 và i là duy nhất, với mọi i. Bob tự hỏi, có bao nhiêu sơ đồ khác nhau có thể có, thoả mãn thời gian truyền tin như anh đã đo được. Hai sơ đồ được coi là khác nhau nếu tồn tại một cạnh có trong sơ đồ này nhưng không có trong sơ đồ kia.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n m.
- Dòng tiếp theo chứa n số nguyên, số thứ i là thời gian truyền tin từ nút 1 đến nút i.

Kết quả

Ghi một số nguyên duy nhất là số sơ đồ khác nhau có thể có, sau khi chia lấy dư cho 1000000007.

Ví dụ

stdin	stdout
4 4 0 1 1 2	2

Hạn chế

- Trong tất cả các test: $1 \leq n, m \leq 10^5$;
- Có 8% số test với $n, m \leq 8$;
- Có 20% số test với $n \leq 100$;
- Có 28% số test với $m = n - 1$;
- Có 44% số test với ràng buộc gốc.

Bài C. prigame

Có 128 tù nhân được đánh số (bằng cách ghi lên áo) từ 1 đến 128. Vào cuối mỗi ngày, họ sẽ cởi áo đưa cho người ở phòng giặt ủi. Sau khi được giặt sạch, các áo được cho vào 128 chiếc hộp kín, được đánh số từ 1 đến 128. Mỗi hộp đựng đúng một chiếc áo, được cho vào một cách **ngẫu nhiên**.

Để ân xá cho các tù nhân khi họ thông minh, quản tù đã đưa ra một thách thức trong ngày: Từng người một được gọi vào phòng giặt ủi theo thứ tự từ 1 đến 128. Mỗi người, khi được gọi vào, sẽ được mở tối đa 64 hộp để nhìn (sau đó đóng lại ngay). Sau đó anh ta trả lời hộp nào chứa áo của mình. Nếu tất cả đều trả lời đúng thì phe tù nhân chiến thắng và tất cả họ được thả tự do. Ngược lại, họ phải chơi lại vào ngày mai.

Yêu cầu: Hãy giúp phe tù nhân đưa ra chiến lược chơi để giành chiến thắng trong thời hạn một tháng (tức là tối đa 30 ngày chơi).

Hệ thống cung cấp thư viện `prisonlib.h` chứa các hàm sau:

```
int watch(int i)
```

dùng để mở hộp thứ i và nhìn vào bên trong. Hàm trả về số được ghi trên chiếc áo trong hộp.

Cần cài đặt hàm:

```
int play(int i)
```

dùng để thực thi chiến thuật chơi của người thứ i. Hàm cần trả về chỉ số của chiếc hộp chứa áo của người này.

Bài D. QSNAIL

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

Một cửa hàng nọ có một thanh gỗ kích thước $1 \times n$ dùng để nuôi ốc sên. Người ta đã dùng bút đánh dấu, chia thanh gỗ thành n ô vuông có độ dài cạnh bằng nhau; được đánh số từ 1 đến n từ trái sang phải. Do thanh gỗ không đều, các ô có thể có mức độ phù hợp khác nhau cho ốc sên; độ phù hợp của ô thứ i là a_i. Một số con ốc sên đã được đặt lên thanh gỗ, thoả mãn mỗi con ốc sên nằm gọn vào một ô và mỗi ô đều chứa không quá một con ốc sên.

Cuội đến cửa hàng để mua ốc sên về nuôi. Cậu dự định phương án là mua các ô từ L đến R và các con ốc sên ở trên đó. Vì Cuội muốn nuôi riêng từng con ốc nên chủ cửa hàng sẽ phải cắt phần gỗ được chọn thành một số đoạn, sao cho trên mỗi đoạn có đúng một con ốc sên. Cụ thể hơn, giả sử có k con ốc sên trên đoạn $[L, R]$, chủ cửa hàng cần cắt đoạn $[L, R]$ thành đúng k đoạn sao cho mỗi đoạn có đúng một con ốc sên và không có đoạn nào của $[L, R]$ bị thừa ra. Vì muốn ốc phát triển tốt, một con ốc sên (trong số vừa được chọn ra) sẽ chỉ được bán nếu tổng độ phù hợp của các ô của đoạn mà nó đứng là không âm. Tức là, con ốc trên đoạn $[u, v]$ được bán nếu $a_u + a_{u+1} + \ldots + a_v \geq 0$. Do có thể có nhiều cách cắt khác nhau, chủ cửa hàng muốn cắt sao cho bán được nhiều ốc sên nhất có thể.

Yêu cầu: Cuội vừa đưa ra Q dự định mua, dự định thứ i được mô tả bởi hai số nguyên dương L_i, R_i. Hãy giúp chủ cửa hàng tìm cách cắt cho từng dự định, sao cho số lượng ốc sên bán được cho dự định đó là lớn nhất có thể và in ra số lượng đó. Lưu ý Cuội chỉ đưa ra các dự định và chủ cửa hàng đưa ra giải pháp chứ chưa thực sự cắt thanh gỗ ban đầu.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n, Q ($n, Q \leq 5 \times 10^5$).
- Dòng tiếp theo chứa n số nguyên $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$).
- Dòng tiếp theo chứa một xâu nhị phân độ dài n, ký tự thứ i là 0 hoặc 1 tương ứng là ô thứ i không đặt hoặc có đặt một con ốc sên.
- Dòng thứ i trong số Q dòng tiếp theo chứa hai số nguyên dương L_i, R_i ($L_i \leq R_i \leq n$).

Dữ liệu bảo đảm có ít nhất một con ốc sên trong đoạn $[L_i, R_i], i = 1, 2, \ldots, Q$.

Kết quả

Gồm Q dòng, dòng thứ i ghi một số nguyên duy nhất là số ốc sên tối đa bán được với dự định thứ i của Cuội.

Ví dụ

stdin	stdout
8 5 1 -2 1 2 -1 2 1 -3 01001101 1 8 1 5 2 5 5 8 1 2	3 2 1 1 0

Hạn chế

- Có 12% số test với $n \leq 5000$ và $Q = 1$.
- Có 12% số test với $n, Q \leq 5000$.
- Có 16% số test với $Q = 1$.

- Có 28% số test với $n, Q \leq 50000$
- Có 32% số test với ràng buộc gốc.

Bài E. HTMAX

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

Cho dãy số nguyên a. Dãy con của a là dãy thu được bằng cách xóa đi một số phần tử của a (có thể không xóa phần tử nào, cũng có thể xóa hết tất cả). Một dãy con được gọi là lỗ chỗ nếu nó không chứa hai phần tử liên tiếp trong a. Hãy tìm dãy con lỗ chỗ có tổng lớn nhất của dãy a

Dữ liệu vào

- Dòng đầu chứa số phần tử của dãy a: n
- Dòng tiếp theo chứa dãy a

Kết quả

- Dòng đầu chứa tổng lớn nhất tìm được
- Dòng tiếp theo chứa k là độ dài dãy con tìm được
- Dòng tiếp theo chứa k số là chỉ số của các phần tử được chọn theo thứ tự trên dãy a

Nếu có nhiều dãy con tốt nhất, hãy in ra dãy con có thứ tự từ điển nhỏ nhất

Ví dụ

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

Hạn chế

- $n \leq 10^5$. $-10^9 \leq a_i \leq 10^9$
- 30% test với $n \leq 20$
- 30% test với $20 < n \leq 1000$

Bài F. BUYGOOD

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

Có m cửa hàng cùng bán một loại sản phẩm, cửa hàng thứ i có s_i sản phẩm giá mỗi sản phẩm là p_i và phải trả w_i tiền hóa đơn nếu mua sản phẩm của cửa hàng đó. Có nghĩa là nếu mua k sản phẩm của cửa hàng thứ i ($k \leq s_i$), bạn phải trả: 0 tiền nếu $k = 0$ hoặc $w_i + k \times p_i$ tiền nếu $k \neq 0$

Yêu cầu: Tính số tiền ít nhất để mua được n sản phẩm

Dữ liệu vào

- Dòng đầu tiên chứa n m
- m dòng tiếp theo mỗi dòng chứa 3 số s_i p_i w_i

Kết quả

- Số tiền ít nhất để mua được n sản phẩm

Ví dụ

stdin	stdout
20 4 5 5 6 10 4 12 15 6 9 20 7 0	118

Hạn chế

- $n \leq 10^4$, $m \leq 100$, $0 \leq si$, $pi \leq 10^4$, $0 \leq w_i \leq 10^6$

Bài G. FPRIME

Một số nguyên dương được gọi là đẹp-nguyên-tố nếu tổng các chữ số của nó là một số nguyên tố; đồng thời nó không chứa chữ số 3 (ở đây đang xét biểu diễn hệ cơ số 10). Cụ thể hơn, giả sử biểu diễn thập phân của số nguyên dương x là $x = x_1x_2 \dots x_l$; khi đó số x được gọi là đẹp-nguyên-tố nếu thoả mãn tất cả các điều kiện sau:

- $x_i \neq 3$ với mọi $i = 1, 2, 3, \dots, l$;
- $x_1 + x_2 + \dots + x_l$ là một số nguyên tố.

Ví dụ 2, 21, 515 là các số đẹp-nguyên-tố; nhưng 31234, 51, 1 đều không phải. Hãy đếm số lượng số đẹp-nguyên-tố trong phạm vi $[L, H]$.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($T \leq 10^5$);
- Mỗi dòng trong số T dòng tiếp theo chứa hai số nguyên dương L, H mô tả một testcase ($L \leq H \leq 10^{1000}$).

Tổng số chữ số của tất cả các số H ở dạng thập phân là không quá 10^6.

Kết quả

Ghi T dòng là kết quả cho T testcase. Do kết quả có thể rất lớn, chỉ cần in ra phần dư khi chia cho 1000000007.

Ví dụ

stdin	stdout
3 1 100 43213 43213 12345 23456	29 0 2004

Hạn chế

- Có 12% số test: $H \leq 10^6$.
- Có 12% số test khác: $T = 1$ và $H \leq 10^9$.
- Có 20% số test khác: $H \leq 10^{18}$.
- Có 24% số test khác: $T = 1$.
- 32% số test còn lại: ràng buộc gốc.

Bài H. BLINDSORT

Đây là một bài toán tương tác (interactive). Hùng được giao nhiệm vụ sắp xếp lại sách trong một thư viện theo độ dày của sách. Có n quyển sách với chiều dày khác nhau đôi một. Có n ô đựng sách, được đánh số từ 1 đến n, mỗi ô đựng đúng một quyển. Tiếc thay, khi nhận nhiệm vụ, Hùng đang không có mặt ở đó; nên cậu sẽ trao đổi thư từ với nhân viên ở thư viện để thực hiện nhiệm vụ của mình. Mỗi lần gửi thư, Hùng sẽ gửi một danh sách gồm các cặp chỉ số: $(i_1, j_1); (i_2, j_2); (i_3, j_3); \dots (i_k, j_k)$ với ý định so sánh độ dày của các cặp quyển sách ở các ô đó. Độ dài của lá thư là k phải không quá $n/2$, và mỗi chỉ số xuất hiện nhiều nhất một lần trong một lá thư. Tức là các số $i_1, j_1, i_2, j_2, \dots, i_k, j_k$ phải đôi một phân biệt. Tuy nhiên, thư viện có hai nhân viên (tạm gọi là A và B) và mỗi người lại hiểu mệnh lệnh theo một cách khác nhau. Nhân viên A xem thư trước, đọc lần lượt từng cặp. Khi đọc đến một cặp trong danh sách, anh ta được lựa chọn là sẽ đổi chỗ hai cuốn sách trong cùng cặp đó, hoặc là không. Có nghĩa là, anh ta sẽ chọn ra một số cặp (i_t, j_t) nào đó, rồi đổi chỗ quyển sách ở ô i_t cho quyển sách ở ô j_t. Việc chọn những cặp nào để đổi, những cặp nào giữ nguyên là hoàn toàn do nhân viên A quyết định, không theo một quy tắc cụ thể nào. Lưu ý là nội dung của lá thư không bị A thay đổi, A chỉ đổi chỗ sách trên kệ mà thôi. Sau khi A hết ca, sẽ tới ca làm của B. Nhân viên B đọc thư và tiến hành so sánh độ dày các cặp quyển sách, sau đó gửi kết quả cho Hùng. Cứ mỗi cặp (i_t, j_t), B sẽ tới nhìn vào ô i_t và j_t rồi ghi lại 1 hoặc 0 tương ứng là quyển sách ở ô i_t dày hơn hay mỏng hơn quyển sách ở ô j_t.

Nhiệm vụ của Hùng là gửi không quá Q lá thư tới thư viện, để cuối cùng anh có thể đưa ra một thứ tự các ô sao cho độ dày các quyển sách theo thứ tự đó là giảm dần.

Tương tác:

Để bắt đầu tương tác, bạn cần đọc vào 2 số nguyên : n, Q. Quá trình tương tác diễn ra như sau:

- Để gửi thư, bạn cần in ra theo cú pháp: "ASK" k i_1 j_1 i_2 j_2 $\dots$ i_k j_k; Sau đó đọc vào một xâu nhị phân độ dài k là kết quả mà B gửi lại. Bit thứ t là 1 cho biết quyển sách ở ô i_t dày hơn quyển sách ở ô j_t; bit 0 trong trường hợp ngược lại.
- Khi tìm ra thứ tự, bạn cần in ra theo cú pháp: "ANSWER" i_1 i_2 $\dots$ i_k là chỉ số của các ô mà theo đó, độ dày các quyển sách là giảm dần.

Sau khi in ra một truy vấn hoặc đáp án, đừng quên xuống dòng và flush đầu ra chuẩn, nếu không bạn có thể nhận verdict `Time limit exceeded`. Để làm điều này, bạn có thể sử dụng:

- `fflush(stdout)` hoặc `cout.flush()` trong C++;
- `System.out.flush()` trong Java;
- `flush(output)` trong Pascal;
- `stdout.flush()` trong Python;
- xem tài liệu chuẩn đối với các ngôn ngữ khác.

Ví dụ

stdin	stdout
4 50 10 0	ASK 2 1 2 3 4 ASK 1 1 4 ANSWER 4 3 1 2

Giải thích

Dãy độ dày ban đầu có thể là: 1, 2, 3, 4. Test VD chỉ mô tả cách thức tương tác, không mô tả thuật toán giải.

Hạn chế

- Trong tất cả các test: $1 \le n \le 500$; $500 \le Q \le 5000$;
- Subtask 1: $Q \ge n^2$.

- Subtask 2: A sẽ không bao giờ đổi chỗ hai cuốn sách; và $Q \geq n \log n$.
- Subtask 3: $Q \geq n \log n$.
- Subtask 4: A sẽ đổi chỗ để cuốn sách dày hơn nằm ở i_t, với mọi t.
- Subtask 5: không có ràng buộc gì thêm.

Bài I. SHIPPER

Một người giao hàng có nhiệm vụ phải vận chuyển kẹo theo đơn đặt hàng để phục vụ dịp trung thu. Bản đồ có thể được hiểu là trục số, tất cả kẹo đều ở vị trí 0. Mỗi đơn hàng có dạng (x, a) - cần giao a kẹo đến vị trí x

Người giao hàng xuất phát từ 0, vì giới hạn riêng mà tại mọi thời điểm, anh ta không thể mang trên người quá k kẹo. Sau khi giao hàng xong, anh ta phải quay về 0 để nộp hóa đơn, hãy tính xem anh ta phải đi ít nhất bao nhiêu để hoàn thành công việc. Biết rằng tất cả kẹo là như nhau và có đủ kẹo ở 0 để giao cho tất cả các đơn hàng

Dữ liệu vào

- Dòng đầu tiên: n k với n là số đơn hàng
- n dòng tiếp theo mỗi dòng mô tả một đơn hàng: (x, a)

Kết quả

Một số nguyên là kết quả bài toán

Ví dụ

stdin	stdout
3 10 0 5 1 2 1 3	2
1 1 -1 10	20

Hạn chế

- $1 \le n \le 10^5$, $-10^6 \le x \le 10^6$, $1 \le k, a \le 10^6$
- Có 50% số test với $n \le 1000$

Bài J. MANT2

Tý có n con kiến, được đánh số từ 1 đến n. Cậu muốn đặt các con kiến lên một sợi dây phơi quần áo, con thứ i được đặt vào vị trí x_i (xem như dây phơi là trục số), không có hai con kiến nào cùng vị trí. Khi kiến bò và có hai con nào đó chạm mặt nhau, chúng sẽ rơi xuống. Tý biết trước vận tốc bò của từng con kiến, và cậu muốn chọn hướng bò cho từng con sao cho thời gian mà cả n con kiến có mặt trên dây phơi là lâu nhất có thể. Hãy giúp Tý.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n là số con kiến; ($n \leq 10^5$).
- Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên dương x_i, v_i là vị trí và vận tốc của con kiến thứ i; ($x_i, v_i \leq 10^9$).

Kết quả

Ghi một số thực duy nhất là thời gian lâu nhất có thể mà cả n con kiến có mặt trên dây phơi, nếu Tý chọn hướng bò cho các con kiến một cách tối ưu. Nếu thời gian bò là vô hạn, ghi ra -1. Kết quả được chấm đúng nếu sai lệch tương đối so với kết quả của giám khảo không quá 10^-6

Ví dụ

stdin	stdout
4 1 2 2 3 3 4 6 2	1.500000

Hạn chế

- Có 20% số test với $n \leq 20$;
- Có 36% số test với $n \leq 1000$.
- Có 44% số test với ràng buộc gốc.

Bài K. SUMCIR

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

Cho một vòng tròn có n vị trí, được đánh số từ 0 đến $n-1$ theo chiều kim đồng hồ. Ta nói khoảng cách giữa vị trí i và j trên vòng tròn này là $\min(|i-j|, n-|i-j|)$.

Ban đầu, vị trí i được điền số tự nhiên a_i. Thực hiện biến đổi vòng này k lần. Mỗi lần ta thay vòng hiện có bởi một vòng mới, số ghi ở vị trí i của vòng mới bằng tổng tất cả các số ghi ở các vị trí trên vòng cũ có khoảng cách đến i không quá D. Yêu cầu tính toán trạng thái cuối cùng của vòng

Dữ liệu vào

- Dòng đầu tiên chứa: $n\ mod\ D\ k$
- Dòng tiếp theo chứa n số: a_i

Kết quả

- Ghi n số trên vòng sau khi biến đổi theo thứ tự đánh số. Do kết quả có thể rất lớn, chỉ cần in ra phần dư khi chia cho mod

Ví dụ

stdin	stdout
5 3 1 1 1 2 2 1 2	2 2 2 2 1
5 3 1 10 1 2 2 1 2	2 0 0 2 2

Hạn chế

- $1 \le n \le 500$, $1 \le mod, k \le 10^7$, $0 \le D < \frac{n}{2}$

Bài L. TREASURE2

Alice và Bob đang tham gia một trò chơi tìm kho báu. Quản trò chuẩn bị một mê cung có 1026 phòng, mỗi phòng có một bóng đèn đang bật hoặc tắt. Mỗi phòng kề với nhiều nhất 2 phòng khác sao cho có thể di chuyển giữa tất cả các phòng với nhau. Kho báu được chôn ở một phòng nào đó mà *kề với đúng 2 phòng khác*, quản trò dẫn Alice vào phòng này, sau đó Alice được tự do di chuyển giữa các phòng, viết lên sàn, thay đổi trạng thái của một số bóng đèn. Tiếp đến, quản trò đưa Alice ra ngoài và xoá sạch mọi dấu vết cô có thể để lại bên trong, bao gồm cả các số được viết trên sàn. Nhưng quản trò sẽ không tác động đến trạng thái của các bóng đèn đang có. Sau đó, quản trò dẫn Bob vào một phòng nào đó, Bob được tự do di chuyển giữa các phòng, viết lên sàn, thay đổi trạng thái của một số bóng đèn. Nhiệm vụ của Bob là phải dừng chân ở phòng mà kho báu được chôn sau đó đào kho báu lên. Các hành động mà người chơi sẽ thực hiện khi bước vào một phòng sẽ được quy định như sau:

- Hành động viết số lên sàn được hệ thống thực hiện tự động, bắt đầu từ 1 và tăng dần mỗi khi đi tới một phòng chưa được viết số.
- Hành động quan sát được hệ thống thực thi tự động và trả về dưới dạng một vector gồm 3 số nguyên, gọi là *vector thông tin*. Số thứ nhất là 0 hoặc 1 tương ứng là trạng thái bóng đèn trong phòng là tắt hoặc bật. Hai số tiếp theo là số được viết trên sàn của các căn phòng dẫn tới bởi các cửa tương ứng với hai bức tường của phòng hiện tại, số -1 đại diện cho trường hợp bức tường là biên của mê cung, số 0 đại diện cho trường hợp sàn chưa được viết số gì, số nguyên dương đại diện cho trường hợp sàn đã được viết số đó. Do ở trong mê cung bị mất phương hướng hoàn toàn, nên thứ tự được quan sát của các bức tường là tuỳ ý, và có thể không giống nhau qua các lần quan sát.
- Hành động thay đổi trạng thái đèn của phòng được thực hiện bằng cách gọi hàm `flip()` được mô tả ở dưới.
- Hành động kết thúc hành trình được thực hiện bằng cách dùng lệnh `return` để thoát khỏi hàm thực thi của nhân vật (hàm `solveAlice` hoặc `solveBob`).
- Hành động di chuyển sang phòng khác được thực hiện bằng cách gọi hàm `move(i)` được mô tả ở dưới.

Yêu cầu: Hãy lập trình mô tả một chiến thuật chơi giúp Bob kết thúc tại phòng chứa kho báu, tức là phòng mà Alice bắt đầu, sao cho số lần `flip` của Alice là **không quá hai**.

Hệ thống cung cấp thư viện `treasure2lib.h` chứa các hàm sau:

```
vector<int> move(int i)
```

- Hàm này dùng để thực hiện di chuyển nhân vật sang phòng khác, đi qua cánh cửa trên bức tường thứ i theo thứ tự trên vector thông tin hiện tại.
- i là một số nguyên thuộc phạm vi [1,2] xác định cánh cửa sẽ đi qua theo thứ tự trên vector thông tin hiện tại.
- Hàm trả về vector thông tin của phòng vừa mới di chuyển vào.

```
void flip()
```

- Hàm này dùng để thay đổi trạng thái của bóng đèn ở phòng mà nhân vật đang đứng.

Thí sinh cần cài hai hàm sau:

```
void solveAlice(vector<int> s)
```

dùng để thực thi chiến thuật chơi của Alice. s là vector thông tin của phòng mà Alice xuất phát.

```
void solveBob(vector<int> s)
```

dùng để thực thi chiến thuật chơi của Bob. s là vector thông tin của phòng mà Bob xuất phát.

Lưu ý:

- Trình chấm sẽ gọi hàm `solveAlice` trước, sau đó sẽ gọi hàm `solveBob` ở một luồng khác với tài nguyên độc lập với luồng chạy `solveAlice`.
- Các hàm có thể được gọi nhiều lần (không quá 5000 lần), tương ứng với nhiều testcase. Thí sinh cần tự khởi tạo lại giá trị các biến nếu cần thiết.
- Hai hàm này cần đặt trong file `treasure2.cpp`.
- Trong file `treasure2.cpp` thí sinh cần khai báo thư viện bằng dòng lệnh `#include "treasure2lib.h"` ở đầu file.

Ngoài ra, thí sinh được phép khai báo thêm thư viện, xây dựng các hàm, sử dụng biến toàn cục khác nếu cần. Tuy nhiên các hàm tự tạo không được phép đặt tên là `main`. File `treasure2.cpp` sẽ được biên dịch cùng với thư viện `treasure2lib.h`.

Đính kèm: Đây là thư viện ví dụ, học sinh có thể sử dụng để kiểm thử bài của mình trước. Hướng dẫn sử dụng có ở đầu của thư viện: https://ideone.com/gYTgsM