

14/11:

Bài A. GGAME

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

Cho một đồ thị có hướng không có chu trình. Đỉnh thứ i của đồ thị có chứa a_i đồng xu. Hai người chơi một trò chơi như sau:

- Hai người chơi luân phiên nhau thực hiện nước đi
- Đến lượt mình, người chơi chọn một đồng xu bất kỳ và di chuyển nó xuống một trong các đỉnh kề của đỉnh đang chứa đồng xu đó. Tất nhiên là không thể chọn đồng xu đang ở nút không thể di chuyển được nữa
- Ai không thực hiện được nước đi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Biết rằng hai người chơi đều rất thông minh, hãy xác định xem liệu người đi trước có chiến thắng hay không

Dữ liệu vào

- Dòng đầu tiên chứa n, m : số đỉnh và số cung của đồ thị
- m dòng tiếp theo mỗi dòng chứa một cung
- Dòng tiếp theo chứa $a_1, a_2, a_3, \dots, a_n$

Kết quả

- Ghi YES/NO tương ứng là người đi trước thắng/thua

Ví dụ

stdin	stdout
5 4 1 2 2 3 3 4 4 5 5 4 3 2 1	NO

Hạn chế

- $0 \leq n, m, a_i \leq 10^5$

Bài B. BOX2

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

Cho n chiếc hộp được đánh số từ 1 đến n . Hộp thứ i có chiều dài a_i , chiều rộng b_i . Hộp i có thể đặt vào trong hộp j nếu i chưa bị chứa bởi hộp nào khác, j đang không chứa hộp nào khác và $a_i < a_j$, $b_i < b_j$. Cần tìm cách lồng các hộp vào nhau sao cho số hộp không bị lồng vào bất kỳ hộp nào là ít nhất. Nếu có nhiều cách lồng các hộp đều là tốt nhất, in ra cách bất kỳ

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n
- Dòng thứ i trong số n dòng tiếp theo ghi a_i b_i

Kết quả

- Dòng đầu chứa số nguyên dương k là số hộp còn lại
- k dòng tiếp theo, mỗi dòng mô tả một dãy hộp: Số đầu tiên là số lượng hộp bị lồng vào nhau và theo sau là chỉ số của các hộp bị lồng vào nhau, theo thứ tự từ ngoài vào trong của các hộp

Các dãy hộp có thể in ra theo thứ tự tùy ý

Ví dụ

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

Hạn chế

- $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$
- Subtask 1: $n \leq 5000$
- Subtask 2: $a_i = b_i$
- Subtask 3: Ràng buộc gốc

Bài C. 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 D. TREEXOR

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

Cho một cây n đỉnh, các đỉnh được đánh số từ 1 đến n . Đỉnh thứ i có trọng số a_i . Cây được gọi là đẹp nếu xor của trọng số của tất cả các đỉnh của cây bằng một số s cho trước. Hãy đếm số cách xoá đi một số cạnh của cây, sao cho sau khi xoá đi thì tất cả các cây được tách ra đều đẹp.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n và số nguyên không âm x .
- Dòng tiếp theo chứa n số nguyên không âm a_1, a_2, \dots, a_n .
- Mỗi dòng trong số $n - 1$ dòng tiếp theo chứa hai số nguyên mô tả một cạnh của cây.

Kết quả

Ghi một số nguyên duy nhất là số cách xoá đi một số cạnh của cây sao cho các cây thu được đều đẹp. Do kết quả có thể rất lớn, chỉ cần in ra phần dư khi chia cho 1000000007.

Ví dụ

stdin	stdout
6 0 0 1 1 0 1 1 1 2 1 3 2 4 2 5 3 6	8

Hạn chế

- Trong tất cả các test: $n \leq 10^5$; $a_i, x \leq 10^9$.
- Có 20% số test với $n \leq 1000$; $a_i \leq 1$ và $x = 0$.
- Có 20% số test với $a_i \leq 1$ và $x = 0$.
- Có 30% số test với $x = 0$.
- Có 30% số test với ràng buộc gốc.

Bài E. CHOCOLATEGAME

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

An và Bình có một miếng socola hình chữ nhật, kích thước $n \times m$. Trên miếng socola có $n - 1$ đường kẻ ngang và $m - 1$ đường kẻ dọc, kết hợp với các đường biên chia hình chữ nhật thành một lưới tọa độ. Các điểm được đánh tọa độ từ $(0, 0)$ đến (n, m) từ trên xuống dưới, từ trái qua phải. Hai người họ sẽ chơi một trò chơi trên đó như sau:

- Hai người luân phiên nhau thực hiện lượt chơi
- Ở lượt chơi của mình, người chơi sẽ chọn ra một miếng socola và cắt đôi nó. Lát cắt phải là một đường thẳng song song với cạnh của hình chữ nhật và ở tọa độ nguyên.
- Ai không thực hiện được lượt chơi hợp lệ sẽ thua cuộc

Hiện tại hai người họ đã chơi được k lần cắt, giờ đến lượt An đi. Hãy tính xem An có bao nhiêu cách thực hiện lượt chơi này (lần cắt thứ $k + 1$) để đảm bảo chiến thắng.

Dữ liệu vào

- Dòng đầu chứa ba số nguyên dương là kích thước ban đầu và số lát cắt: $n \ m \ k$ ($1 \leq n, m \leq 10^9, 1 \leq k \leq 10^5$)
- k dòng tiếp theo, mỗi dòng ghi một lát cắt: $u \ v \ x \ y$ ($0 \leq u \leq x \leq n, 0 \leq v \leq y \leq m$) có nghĩa lát cắt này kéo dài từ điểm (u, v) đến điểm (x, y) . Dữ liệu đảm bảo lát cắt song song với biên, không trùng lên biên của hình chữ nhật đang cắt, có độ dài khác 0 và là một lượt chơi hợp lệ theo đúng thứ tự chơi.

Kết quả

Ghi một số nguyên không âm là số cách chơi lượt tiếp theo của An để đảm bảo chiến thắng.

Ví dụ

stdin	stdout
1 5 1 0 2 1 2	3

Hạn chế

- Subtask 1 (30%): $n, m \leq 100$
- Subtask 2 (30%): $k \leq 1000$
- Subtask 3 (40%): Ràng buộc gốc

Bài F. WORKSMART

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

Doanh nghiệp xây dựng của Hùng hiện đang nhận được thông tin của n công trình cần xây dựng, được đánh số từ 1 đến n . Lợi nhuận mà công trình thứ i sẽ mang lại là b_i (có thể là số âm thể hiện thua lỗ). Một số công trình có thể có yêu cầu về kinh nghiệm xây dựng, theo đó doanh nghiệp của Hùng cần xây dựng và hoàn thành công trình q_i trước khi có thể bắt đầu công trình thứ i .

Doanh nghiệp hiện tại có số vốn là v , và có thể quyết định chọn các công trình nào để xây dựng và chọn thứ tự để xây dựng chúng, miễn là các điều kiện về kinh nghiệm phải được thỏa mãn. Ngoài ra, số tiền còn lại của doanh nghiệp không được phép âm tại bất kỳ thời điểm nào.

Yêu cầu: Hãy tính lợi nhuận tối đa mà doanh nghiệp của Hùng có thể đạt được (có thể là không làm bất cứ công trình nào).

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên n và v — số lượng công trình và số tiền vốn hiện tại.
- n dòng tiếp theo, dòng thứ i chứa hai số nguyên b_i và q_i — lợi nhuận và yêu cầu kinh nghiệm của công trình thứ i . Nếu $q_i = 0$ cho biết công trình thứ i không có yêu cầu về kinh nghiệm.

Kết quả

In ra một số nguyên duy nhất là lợi nhuận tối đa có thể đạt được.

Ví dụ

stdin	stdout
5 2 4 0 -2 0 -3 2 -2 1 15 3	14

Giải thích

Có thể chọn các công trình 1, 2, 3, 5 theo đúng thứ tự đó. Tổng vốn cuối cùng là $2+4-2-3+15=16$, do đó lợi nhuận là $16-2=14$.

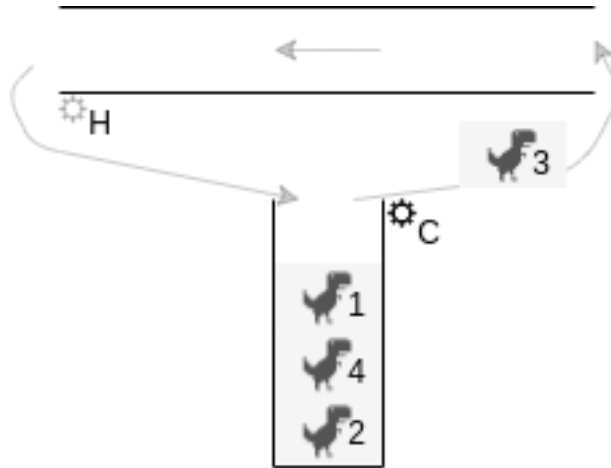
Hạn chế

- Trong tất cả các test: $1 \leq n \leq 3 \times 10^5$; $0 \leq v \leq 10^9$; $-10^9 \leq b_i \leq 10^9$; $0 \leq q_i < i$.
- Subtask 1:** $v = 10^{18}$.
- Subtask 2:** $n \leq 2000$, và với mỗi i thì $q_i = i - 1$ hoặc $q_i = 0$.
- Subtask 3:** với mỗi i thì $q_i = i - 1$ hoặc $q_i = 0$.
- Subtask 4:** $n \leq 2000$.
- Subtask 5:** không có ràng buộc gì thêm.

Bài G. SDINO

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1.5 giây
Hạn chế bộ nhớ: 512MB

Khủng long kỳ thực không hề hung dữ như người ta vẫn tưởng, chúng rất thông minh và dễ bảo. Bạn là người trông giữ khủng long và có nhiệm vụ sắp xếp lại khủng long trong chuồng. Chuồng chỉ có một cửa ra vào, và chiều ngang hẹp chỉ vừa đủ cho một con khủng long di chuyển, vì thế con nào vào chuồng trước thì sẽ phải ra sau. Ở cạnh chuồng có một hành lang. Hành lang có một cửa vào và một cửa ra, và chiều ngang cũng vừa đủ cho một con khủng long di chuyển, nên con nào vào trước sẽ phải ra trước.



Có n con khủng long, tất cả đều đang ở trong chuồng. Con khủng long thứ i (tính từ cửa chuồng vào trong) có số hiệu p_i (p_1, p_2, \dots, p_n là một hoán vị của $\{1, 2, \dots, n\}$). Để sắp xếp lại các con khủng long, bạn sẽ sử dụng đèn báo hiệu được lắp ở chuồng và hành lang. Khi bật đèn báo hiệu ở chuồng, nếu trong chuồng có khủng long thì một con khủng long trong chuồng sẽ đi sang hành lang. Khi bật đèn báo hiệu ở hành lang, nếu ở hành lang có khủng long thì một con khủng long ở hành lang sẽ đi vào chuồng.

Yêu cầu: Hãy chỉ ra một dãy không quá 10^6 báo hiệu để sắp xếp lại các con khủng long theo thứ tự từ nhỏ đến lớn tính từ cửa chuồng vào trong.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n
- Dòng tiếp theo chứa n số nguyên dương $p_1 p_2 \dots p_n$

Kết quả

Ghi chuỗi s gồm nhiều ký tự viết liền nhau, các ký tự C cho biết bạn bật đèn ở chuồng, các ký tự H cho biết bạn bật đèn ở hành lang. Lưu ý các đèn là đèn báo hiệu và sẽ tắt ngay sau khi được bật sáng. Các đèn được bật theo đúng trình tự trong chuỗi s . Nếu s là một dãy điều khiển thoả mãn, bạn sẽ nhận được $\min(1, \frac{10^6}{|s|})$ tổng điểm của test đó.

Ví dụ

stdin	stdout
4 3 1 4 2	CCHHCCCCCHHHHCCHHCCHCCHCHHHH

Hạn chế

- Có 12% số test với $n \leq 50$;
- Có 24% số test với $n \leq 1000$;
- Có 64% số test với $n \leq 20000$.

Bài H. DEGDAG

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

Một đồ thị có hướng G được gọi là đồ thị nghịch thế nếu tồn tại một dãy số thực phân biệt $x = x_1, x_2, \dots, x_n$ sao cho: G có n đỉnh được đánh số từ 1 đến n ; có một cung nối từ đỉnh i đến đỉnh j khi và chỉ khi $i < j$ và $x_i > x_j$. Dãy đặc trưng của G là $(deg^+[1], deg^+[2], \dots, deg^+[n])$ với $deg^+[i]$ là số cung đi vào đỉnh i . Số lượng đỉnh có bậc ra bằng 0 (tức là không có cung đi ra) được gọi là độ-đa-dạng-kết-cục của G . Cho biết dãy đặc trưng của G , hãy xây dựng đồ thị G sao cho độ-đa-dạng-kết-cục của nó là lớn nhất có thể.

Vì sợ bài này chưa đủ khó, Hùng đã thêm vào các query. Cậu bắt đầu với một dãy số nguyên không âm a_1, a_2, \dots, a_m . Cậu sẽ thực hiện Q thao tác thuộc một trong hai loại:

- $+ L H d$: Gán $a_i = a_i + d$ với mọi $i = L, L+1, \dots, H$.
- $? L H$: Dựng đồ thị có dãy đặc trưng là $(deg^+[1] = a_L, deg^+[2] = a_{L+1}, \dots, deg^+[H-L+1] = a_H)$ sao cho độ-đa-dạng-kết-cục của nó là lớn nhất có thể. Sau đó in ra độ-đa-dạng-kết-cục của nó.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên: m và Q .
- Dòng tiếp theo chứa m số nguyên không âm: a_1, a_2, \dots, a_m .
- Mỗi dòng trong Q dòng tiếp theo chứa một truy vấn như mô tả.

Kết quả

Với mỗi thao tác loại $?$, in ra độ-đa-dạng-kết-cục của đồ thị đã xây dựng được; hoặc in -1 nếu không tồn tại đồ thị nào thoả mãn.

Ví dụ

stdin	stdout
7 3	5
0 0 1 0 2 1 0	4
? 1 7	
+ 6 7 1	
? 2 7	

Giải thích

Dãy x cho đồ thị ở truy vấn 1 có thể là: $(1, 4, 2, 6, 3, 5, 7)$ với các đỉnh không có cung đi ra là 1, 3, 5, 6, 7
Dãy x cho đồ thị ở truy vấn 3 có thể là: $(4, 1, 6, 2, 3, 5)$ với các đỉnh không có cung đi ra là 2, 4, 5, 6

Hạn chế

- Trong tất cả các test: $m, Q \leq 2 \times 10^5$; $-10^9 \leq w \leq 10^9$.
- Có 24% số test với $m, Q \leq 9$.
- Có 24% số test với $m, Q \leq 5000$.
- Có 24% số test không có truy vấn loại $+$.
- Có 28% số test không có ràng buộc gì thêm.

15/11:

Bài I. XGUN

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

Bối cảnh trò chơi XGUN là một nửa quả đồi, được chia thành n cột. Độ cao của cột i là h_i ($0 \leq h_1 \leq h_2 \leq \dots \leq h_n \leq 10^9$). Trò chơi diễn ra m lượt theo trình tự, mỗi lượt được đặc trưng bởi hai số i, L có nghĩa là nhân vật ở cột i bắn một tia độ dài L về bên phải. Khi nhân vật ở cột i bắn một tia độ dài L về bên phải, các cột $i+1, i+2, \dots, i+L$ sẽ giảm độ cao về bằng với độ cao hiện tại của cột i . Điểm của lượt chơi này là số lượng đất bị phá, tức là tổng mức giảm độ cao của các cột bị giảm.

Yêu cầu: Hãy tính điểm của từng lượt chơi.

Dữ liệu vào

- Dòng đầu chứa n
- Dòng tiếp theo chứa $h_1 \ h_2 \ \dots \ h_n$
- Dòng tiếp theo chứa số lượt chơi m
- m dòng tiếp theo mỗi dòng chứa một lượt chơi: $i \ L$ ($1 \leq L \leq n - i$)

Kết quả

In ra trên m dòng là điểm của m lượt chơi

Ví dụ

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

Hạn chế

- $1 \leq n, m \leq 10^5, 0 \leq h_i \leq 10^9$
- Có ít nhất 50% số test với $n, m \leq 5000$

Bài J. FLIGHT

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

Cho n bóng đèn và m công tắc. Đèn thứ i đang ở trạng thái a_i và được mắc với 2 công tắc x_i và y_i . Trạng thái đèn là 0/1, còn công tắc là nút ấn - cái mà khi ấn sẽ thay đổi trạng thái của tất cả đèn kết nối với nó

Hãy chỉ ra một cách ấn các công tắc bất kỳ, sao cho tất cả đèn đều bị tắt (0) và không có công tắc nào bị ấn nhiều hơn 1 lần

Dữ liệu vào

- Dòng đầu tiên chứa n m
- n dòng tiếp theo, dòng i chứa a_i x_i y_i

Kết quả

- Nếu không thể tắt, in ra -1
- Ngược lại, dòng đầu tiên in ra số thao tác, dòng thứ 2 in ra dãy các công tắc theo thứ tự sẽ ấn

Ví dụ

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

Hạn chế

- $1 \leq n, m \leq 10^5$
- 20% test có $n, m \leq 5000$

Bài K. 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"
```

Bài L. CPCNT

Hạn chế thời gian: 1 giây

Hãy đếm số cặp (i, j) nguyên tố cùng nhau, sao cho $1 \leq i \leq n$ và $1 \leq j \leq m$. Hai số nguyên tố cùng nhau nếu ước chung lớn nhất của chúng bằng 1.

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: $n \ m$ ($n, m \leq 10^5$).

Kết quả

In ra T dòng là số cặp nguyên tố cùng nhau tìm được tương ứng cho T testcase.

Ví dụ

stdin	stdout
2	9
3 4	63
10 10	

Hạn chế

- Subtask 1: $n, m \leq 1000$;
- Subtask 2: $n = m$;
- Subtask 3: Ràng buộc gốc.

Bài M. SUMSQUARE

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

Cho hai số tự nhiên n, M
Tính $S = 1^2 + 3^2 + 5^2 + \dots + (2n+1)^2$. Chỉ cần đưa ra phần dư khi chia S cho M

Dữ liệu vào

Ghi hai số nguyên dương: n M

Kết quả

Ghi kết quả tính được (tức $S \% M$)

Ví dụ

<code>stdin</code>	<code>stdout</code>
10 1000	771

Hạn chế

- Trong tất cả các test: $0 \leq n, M \leq 10^{18}$, $A \leq B$
- Có 50% số test với $n, M \leq 1000$
- Có 30% số test với $1000 < n, M \leq 10^9$

Bài N. BINACOUNT

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

Cho một xâu S chỉ chứa các ký tự a, b . Có thể biến đổi xâu 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 xâu nhị phân (xâu chỉ chứa các ký tự 0 hoặc 1) khác nhau có thể tạo ra.

Dữ liệu vào

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

Kết quả

Ghi một số nguyên duy nhất là số lượng xâu 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 O. WATER

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

Chỗ ở của Đỗ Nam Trung (nhà xanh) có dạng một hình tròn. Có n cây xanh được trồng xung quanh đó, để thuận tiện các cây sẽ được đánh số từ 1 đến n ngược chiều kim đồng hồ. Tại tâm của hình tròn ông đặt một máy tưới nước, máy này được lập trình để tưới nước cho tất cả các cây. Có m lệnh sẽ được thực hiện tuần tự, lệnh thứ i là tưới các cây theo thứ tự ngược chiều kim đồng hồ, bắt đầu từ cây được đánh số l_i đến cây r_i .

Sau khi đến Việt Nam và tiếp xúc với các coder, Trung nhận ra dãy lệnh của ông có thể dư thừa. Ông muốn xóa bỏ một số nhiều nhất các lệnh sao cho dãy lệnh còn lại vẫn đảm bảo tưới hết cho các cây. Hãy giúp ông!

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n m
- m dòng tiếp theo, dòng thứ i chứa: l_i r_i

Kết quả

- Ghi một số nguyên là số lệnh có thể xóa. In ra -1 nếu Trung không thể tưới hết các cây của mình

Ví dụ

stdin	stdout
6 3 5 6 2 5 4 1	1

Hạn chế

- $1 \leq n, m \leq 10^5$
- Có 30% số test với $m \leq 5000$

Bài P. QUERYK2

File dữ liệu vào: QUERYK2.INP
File kết quả: QUERYK2.OUT
Hạn chế thời gian: 2 giây
Hạn chế bộ nhớ: 512 MB

Hùng có một dãy số nguyên không âm $a = a_1, a_2, \dots, a_n$. Cậu sẽ thực hiện Q truy vấn, mỗi truy vấn thuộc một trong các dạng sau:

- Loại 1: Cho một số nguyên không âm x , chèn thêm một phần tử có giá trị bằng x vào cuối dãy a và tăng giá trị của n lên một đơn vị;
- Loại 2: Xoá đi một phần tử cuối cùng của dãy a và giảm giá trị của n đi một đơn vị;
- Loại 3: Cho một số nguyên không âm x , gán $a_i = a_i \wedge x$ với $i = 1, 2, 3, \dots, n$;
- Loại 4: Cho ba số nguyên L, H, x , tìm $\max(a_i \wedge x)$ với $i = L, L+1, \dots, H$;
- Loại 5: Cho ba số nguyên L, H, x , đếm số lượng phần tử bé hơn hoặc bằng x trong số các phần tử a_L, a_{L+1}, \dots, a_H ;
- Loại 6: Cho ba số nguyên L, H, k , tìm phần tử bé thứ k trong số các phần tử a_L, a_{L+1}, \dots, a_H . Cụ thể hơn, đặt $b_i = a_{i+L-1}$ với $i = 1, 2, \dots, H-L+1$, sắp xếp lại các phần tử của b theo thứ tự tăng dần và đưa ra phần tử thứ k .

Phép toán \wedge được sử dụng trong bài này là phép toán xor. $x \wedge y$ được tính như sau: Biểu diễn x và y dưới dạng nhị phân $x = x_1x_2\dots x_k$ và $y = y_1y_2\dots y_k$ với k là số nguyên dương nhỏ nhất thỏa mãn $2^k > \max(x, y)$. Tính $z_i = (x_i + y_i) \% 2$ với $1 \leq i \leq k$. Khi đó $z = z_1z_2\dots z_k$ chính là kết quả của $x \wedge y$ được biểu diễn dưới dạng nhị phân.

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 không âm a_1, a_2, \dots, a_n ;
- Mỗi dòng trong số Q dòng tiếp theo mô tả một truy vấn theo định dạng:
 - 1 x : Chèn thêm một phần tử có giá trị bằng x vào cuối dãy a và tăng giá trị của n lên một đơn vị ($0 \leq x \leq 5 \times 10^5$);
 - 2: Xoá đi một phần tử cuối cùng của dãy a và giảm giá trị của n đi một đơn vị (dữ liệu đảm bảo $n > 0$ khi thực hiện truy vấn này);
 - 3 x : Gán $a_i = a_i \wedge x$ với $i = 1, 2, 3, \dots, n$ ($0 \leq x \leq 5 \times 10^5$);
 - 4 $L H x$: Tìm $\max(a_i \wedge x)$ với $i = L, L+1, \dots, H$ ($1 \leq L \leq H \leq n, 0 \leq x \leq 5 \times 10^5$);
 - 5 $L H x$: Đếm số lượng phần tử bé hơn hoặc bằng x trong số các phần tử a_L, a_{L+1}, \dots, a_H ($1 \leq L \leq H \leq n, 0 \leq x \leq 5 \times 10^5$);
 - 6 k : Tìm phần tử bé thứ k trong số các phần tử a_L, a_{L+1}, \dots, a_H ($1 \leq k \leq H-L+1$).

Kết quả

Với mỗi truy vấn loại 4, 5 hoặc 6, in ra trên một dòng một số nguyên không âm là kết quả cho truy vấn tương ứng.

Ví dụ

QUERYK2.INP	QUERYK2.OUT
4 6	2
2 4 1 2	3
3 2	4
5 1 4 1	
2	
1 3	
6 2 4 2	
4 1 3 2	

Hạn chế

- Subtask 1 (4%): $n, Q \leq 1000$.
- Subtask 2 (8%): Không có truy vấn loại 3 và $L = 1, R = n$ trong tất cả các truy vấn loại 4, 5, 6.
- Subtask 3 (12%): Không có truy vấn loại 4 và 6.
- Subtask 4 (24%): Không có truy vấn loại 3.
- Subtask 5 (20%): $L = 1$ trong tất cả các truy vấn loại 4, 5, 6.
- Subtask 6 (32%): Không có ràng buộc gì thêm.