

30/10:

Bài A. DOMISTR

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

Xâu t được gọi là đẹp nếu tồn tại một ký tự xuất hiện trong t với số lần nhiều hơn hẳn một nửa số lượng ký tự của t . Cho chuỗi s và Q truy vấn dạng L, R : Trong số các chuỗi con liên tiếp trên đoạn $[L, R]$ của s , có chuỗi đẹp nào độ dài lớn hơn 1 hay không? Cụ thể, tồn tại hay không hai số i, j sao cho $L \leq i < j \leq R$ sao cho chuỗi s_i, s_{i+1}, \dots, s_j là chuỗi đẹp?

Dữ liệu vào

- Dòng đầu chứa chuỗi s ;
- Dòng tiếp theo chứa số nguyên dương Q ;
- Mỗi dòng trong số Q dòng tiếp theo chứa hai số L, R mô tả một truy vấn.

Kết quả

Ghi Q dòng, mỗi dòng là yes/no tương ứng là câu trả lời cho một truy vấn.

Ví dụ

stdin	stdout
traiheMTTN	no
3	yes
1 5	no
1 10	
5 7	

Hạn chế

- $1 \leq |s|, Q \leq 10^5$;
- 50% test có $n \leq 5000$.

Bài B. 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 C. SUMK

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 3 giây

Trong tiết học thể dục có n học sinh xếp thành một hàng, được đánh số từ 1 đến n từ trái sang phải. Học sinh thứ i có chiều cao là h_i . Ta nói hai học sinh i và j không nhìn thấy nhau nếu ở giữa họ có một người cao hơn cả i lẫn j ; tức là tồn tại k ($i < k < j$ hoặc $j < k < i$) sao cho $h_k > h_i$ và $h_k > h_j$. Thầy giáo đang có m viên bi, thầy sẽ chọn ra một số bạn để bắt đầu một trò chơi với các viên bi này. Cách chọn của thầy là hợp lệ nếu những bạn được chọn đôi một không nhìn thấy nhau, và tổng chiều cao của họ là bé hơn hoặc bằng m (sau đó, thầy đưa cho mỗi bạn số bi bằng đúng chiều cao của bạn đó và bắt đầu chơi). Thầy giáo bối rối vì có quá nhiều cách chọn hợp lệ, hãy giúp thầy tính toán con số này. Cụ thể, hãy đếm xem có bao nhiêu cách chọn ra một số học sinh sao cho những bạn được chọn đôi một không nhìn thấy nhau và tổng chiều cao của những bạn được chọn là bé hơn hoặc bằng m . Hai cách chọn được coi là khác nhau nếu tồn tại một học sinh được chọn trong cách này nhưng không được chọn trong cách kia. Lưu ý, không chọn học sinh nào cũng được xem là một cách chọn hợp lệ.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n và m là số học sinh và số bi;
- Dòng thứ hai chứa n số nguyên dương h_1, h_2, \dots, h_n là chiều cao của các học sinh.

Kết quả

Ghi một số nguyên duy nhất là số cách chọn hợp lệ sau khi chia lấy dư cho 1000000007.

Ví dụ

<code>stdin</code>	<code>stdout</code>
6 8 1 6 4 7 5 3	12

Giải thích

Những cách chọn hợp lệ cho test ví dụ là: $\{\}$; $\{1\}$; $\{2\}$; $\{3\}$; $\{4\}$; $\{5\}$; $\{6\}$; $\{1, 3\}$; $\{1, 5\}$; $\{1, 6\}$; $\{3, 6\}$; $\{1, 3, 6\}$.

Hạn chế

- Trong tất cả các test: $n, m \leq 4000$; $h_i \leq m$.
- Có 20% số test với $n \leq 18$ và $m \leq 100$.
- Có 20% số test với $n \leq 36$ và $m \leq 200$.
- Có 28% số test với $n \leq 100$ và $m \leq 400$.
- Có 32% số test không có ràng buộc gì thêm.

Bài D. PINZ

File dữ liệu vào:	standard input
File kết quả:	standard output
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 E. 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

<code>stdin</code>	<code>stdout</code>
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 F. XONDCNT

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

Các cặp đôi yêu nhau thường hay giận dỗi vì sự khác biệt trong quan điểm. Tuy nhiên, có những tình huống nhất định mà cả hai người đều có nhận định giống nhau. Ví dụ, một người thích ăn ngọt còn người kia thích ăn mặn, thì có thể cả hai sẽ cùng thích món sườn xào.

Trong trường hợp nan giải lần này, Th thì thích phép toán `and` còn Tr thì thích phép `xor`. Vì vậy khi đánh giá vẻ đẹp của một dãy số, Th sẽ thấy độ đẹp của dãy này là `and` của các số trong dãy; còn Tr sẽ thấy độ đẹp của nó là `xor` của các số trong dãy. Có rất nhiều dãy số khiến sự khác biệt này thể hiện ra. Tuy nhiên cũng có những dãy số mà cả hai người đều cùng một nhận định.

Cho dãy số nguyên $a = a_1, a_2, \dots, a_n$. Th muốn chọn ra một đoạn con gồm các phần tử liên tiếp của a , sao cho `xor` của các phần tử bằng với `and` của các phần tử. Hãy giúp anh ấy đếm xem có bao nhiêu đoạn con thoả mãn.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n ($n \leq 10^5$);;
- Dòng thứ hai chứa n số nguyên không âm mô tả dãy a ($a_i < 2^{31}$).

Kết quả

Ghi một số nguyên duy nhất là số đoạn con tìm được.

Ví dụ

stdin	stdout
6 3 1 4 2 3 4	7

Hạn chế

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

Bài G. FINARY

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

Hệ cơ số fibonacci được định nghĩa như sau: Đầu tiên tính dãy fibonacci $f_1 = 1, f_2 = 2, f_n = f_{n-1} + f_{n-2}$. Sau đó, mỗi số tự nhiên x sẽ được biểu diễn thành tổng của các số fibonacci bởi thuật toán:

- Nếu $x = 0$ thì dừng;
- Tìm n lớn nhất sao cho $f_n \leq x$, chọn f_n vào tổng;
- Gán $x = x - f_n$, quay lại bước 1.

Cuối cùng, ta dùng một xâu nhị phân để biểu diễn kết quả, bit thứ i là 1 khi và chỉ khi f_i được chọn vào tổng. Ví dụ, các số từ 1 đến 10 sẽ được biểu diễn lần lượt là: 1, 10, 100, 101, 1000, 1001, 1010, 10000, 10001, 10010.

Tiến hành viết biểu diễn của các số từ 1 tới 10^{2023} liên nhau, thu được xâu s . Hãy tính xem trong n ký tự đầu tiên của s có bao nhiêu ký tự là 1.

Dữ liệu vào

Chứa số nguyên n ($0 \leq n \leq 10^{15}$).

Kết quả

Ghi số lượng ký tự 1.

Ví dụ

<code>stdin</code>	<code>stdout</code>
33	14

Giải thích

33 ký tự đầu tiên của s là 110100101100010011010100001000110 và có 14 ký tự 1.

Bài H. MIXCOLOR

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2.0 second
Hạn chế bộ nhớ: 256 MiB

An là một họa sỹ tài năng. Cậu đang muốn tạo ra một màu sắc bằng cách trộn ba màu cơ bản: đỏ, vàng, và lục chính xác theo một tỷ lệ nhất định giữa ba màu này. Ông có một bộ chai chứa các hỗn hợp màu đã được trộn sẵn, với mỗi chai chứa lượng màu đỏ, vàng, và lục khác nhau.

Kệ màu của An không cố định, mà có thể có thêm hoặc bớt đi các chai; khi An mua thêm các chai mới hoặc cho người khác mượn. Vì vậy, An cần tìm cách trộn sau mỗi lần thay đổi bộ chai. An muốn biết liệu có thể trộn được màu sắc yêu thích của mình từ các chai có sẵn hay không, và nếu có — số lượng ít nhất các chai cần dùng để trộn ra màu yêu thích đó là bao nhiêu.

Dữ liệu vào

- Dòng đầu tiên chứa ba số nguyên không âm R_f, Y_f, G_f ($0 \leq R_f, Y_f, G_f$; $0 < R_f + Y_f + G_f \leq 10^6$) mô tả lượng màu đỏ, vàng, và lục trong màu sắc yêu thích của An. Đối với bất kỳ số thực $\alpha > 0$, $(\alpha R_f, \alpha Y_f, \alpha G_f)$ cũng là một hỗn hợp màu yêu thích của An.
- Dòng thứ hai chứa một số nguyên dương Q là số lần thay đổi trên kệ ($Q \leq 10^5$). Ban đầu kệ trống.
- Mỗi dòng trong số Q dòng tiếp theo mô tả một thay đổi trên kệ:
 - Nếu một chai mới được thêm vào, dòng đó sẽ chứa chữ cái in hoa A, theo sau là ba số nguyên không âm R_i, Y_i, G_i ($0 \leq R_i, Y_i, G_i$; $0 < R_i + Y_i + G_i \leq 10^6$) mô tả lượng màu đỏ, vàng, và lục trong chai được thêm vào. Các chai thêm vào được đánh số liên tiếp bằng các số nguyên duy nhất bắt đầu từ 1, tức là chai thứ i tương ứng với chai thứ i được thêm vào trong dữ liệu đầu vào.
 - Nếu một chai bị gỡ bỏ khỏi kệ, dòng đó sẽ chứa chữ cái in hoa R theo sau là một số nguyên r_i . Tất cả các giá trị r_i trong các lần gỡ bỏ là duy nhất, r_i không vượt quá tổng số chai đã thêm cho đến thời điểm đó.

Kết quả

In ra Q dòng. Dòng thứ j ($1 \leq j \leq Q$) chứa số chai ít nhất cần thiết để tạo ra hỗn hợp màu với tỷ lệ yêu thích của An bằng cách sử dụng các chai có sẵn sau j lần thay đổi đầu tiên trên kệ; nếu không thể trộn được thì in ra 0.

Ví dụ

stdin	stdout
3 1 2	0
8	2
A 7 5 6	0
A 17 3 10	1
R 1	0
A 6 2 4	0
R 3	2
A 5 8 5	0
A 1 1 1	
R 2	

Hạn chế

- Subtask 1: $Q \leq 50, 0 < R_i + Y_i + G_i \leq 100$
- Subtask 2: $Q \leq 500, 0 < R_i + Y_i + G_i \leq 1000$
- Subtask 3: $Q \leq 5000, 0 < R_i + Y_i + G_i \leq 10000$
- Subtask 4: Không có ràng buộc gì thêm.

31/10:

Bài I. XEKHACH

Hạn chế thời gian: 1 giây

Đất nước XYZ có n thành phố và $n - 1$ con đường bảo đảm đi lại giữa tất cả các thành phố. Nhà xe PTZ có nhiều chuyến xe khách cố định di chuyển giữa các cặp thành phố. Khi có một khách hàng lên một xe ở bến xuất phát, xe sẽ chở anh ta theo lộ trình định sẵn, tuy nhiên có thể dừng lại dọc đường để trả khách. Do đó, nếu một người muốn đi tới thành phố r và chọn tuyến xe khách đi từ x đến y , anh ta sẽ dừng ở thành phố gần r nhất trên đường đi đơn giữa x và y .

Yêu cầu: Với mỗi yêu cầu của hành khách, hãy tìm thành phố để trả khách.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n .
- Mỗi dòng trong số $n - 1$ dòng tiếp theo chứa hai số nguyên mô tả một cạnh của cây.
- Dòng tiếp theo chứa số nguyên dương Q là số lượng yêu cầu.
- Mỗi dòng trong số Q dòng tiếp theo chứa ba số nguyên dương r, x, y mô tả một yêu cầu của hành khách.

Kết quả

In ra Q dòng là thành phố sẽ trả khách trong Q yêu cầu tương ứng.

Ví dụ

test	answer
6	1
1 2	2
1 3	1
2 4	2
2 5	
3 6	
4	
1 2 3	
4 5 6	
1 4 6	
5 4 6	

Hạn chế

- Trong tất cả các test: $n, Q \leq 2 \times 10^5$;
- Có 20% số test với $n, a_i \leq 1000$.
- Có 30% số test với $r = 1$.

Bài J. GCLEAR2

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2.0 second
Hạn chế bộ nhớ: 256 MB

Robot hút bụi đang di chuyển trong một căn phòng hình chữ nhật, được mô tả bằng bản đồ hình chữ nhật $R \times C$ ô, với “#” đại diện cho tường và “.” đại diện cho sàn nhà. Ví dụ:

```
. . . # # .  
. # . . . .  
# # . # # .  
. . . . # .  
. # . . . .
```

Mỗi phút, robot phát ra tín hiệu cho biết hướng tiếp theo mà nó sẽ di chuyển. Hướng di chuyển luôn là một trong các hướng: Bắc (N), Đông (E), Nam (S), Tây (W)

Bạn có một hệ thống radar để thu thập tín hiệu di chuyển của robot trong M phút, được biểu diễn dưới dạng chuỗi gồm M ký tự, ví dụ “?NE?WS”. Một số tín hiệu có thể không giải mã được và được đánh dấu bằng “?”.

anạn không biết vị trí ban đầu của robot, nhưng bạn cần tính toán xem sau khi thu thập các tín hiệu trên, có bao nhiêu ô trên lưới có thể là vị trí kết thúc của robot, với điều kiện robot luôn di chuyển trên các ô sàn nhà một cách hợp lệ.

Dữ liệu vào

- Dòng đầu tiên chứa ba số nguyên R , C , và M .
- R dòng tiếp theo, mỗi dòng chứa một chuỗi dài C ký tự, mô tả lưới căn phòng gồm các ký tự “#” (tường) và “.” (sàn nhà).
- Dòng cuối cùng chứa chuỗi M ký tự mô tả tín hiệu radar, mỗi ký tự thuộc tập {N, E, S, W, ?}.

Kết quả

In ra một số nguyên duy nhất là số lượng ô khác nhau trên lưới có thể là vị trí kết thúc của robot.

Ví dụ

stdin	stdout
5 6 6 ...##. .#.... #...#.#. .#.... ?NE?WS	5

Hạn chế

- Subtask 1: $1 \leq R, C, M \leq 100$; không có ký tự “?” và R, C, M không vượt quá 100.
- Subtask 2: R, C, M không vượt quá 100.
- Subtask 3: R, C, M không vượt quá 500 và M không vượt quá 5000.

Bài K. FIBOR

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

Dãy fibonacci được định nghĩa như sau:

$$f_1 = 1, f_2 = 2, f_n = f_{n-1} + f_{n-2} \forall n > 2.$$

Có thể thấy rằng bất kỳ số nguyên dương nào cũng có thể biểu diễn thành tổng của các số fibonacci phân biệt, và cách biểu diễn đó là duy nhất. Gọi $S(x)$ là tập các số fibonacci khi biểu diễn x thành tổng của các số fibonacci phân biệt. Số lượng số hạng khi biểu diễn x , tức $|S(x)|$, được gọi là hạng của x . Ví dụ, $S(19) = \{1, 5, 13\}$ nên hạng của 19 là 3.

Trà có hai số nguyên dương x và y với hạng bằng nhau. Cô giữ số lớn hơn (giả sử đó là x) và đưa cho bạn trai số bé hơn. Vì xa nhau lâu ngày nên cô đã quên mất số y . Cô thắc mắc là có bao nhiêu số y có thể.

Yêu cầu: Cho số x dưới dạng biểu diễn bằng $S(x)$, hãy đếm số lượng số y bé hơn x và có hạng bằng x .

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n là số phần tử của tập $S(x)$;
- Dòng tiếp theo chứa n số nguyên dương i_1, i_2, \dots, i_n cho biết các số trong tập $S(x)$ là $f_{i_1}, f_{i_2}, \dots, f_{i_n}$.

Kết quả

Ghi một số nguyên duy nhất là số lượng số y có thể, sau khi chia lấy dư cho 1000000007.

Ví dụ

stdin	stdout
3 1 4 6	2

Giải thích

Ở test ví dụ, $x = f_1 + f_4 + f_6 = 1 + 5 + 13 = 19$. Các số y có thể là: 17, 12.

Hạn chế

- Trong tất cả các test: $1 \leq n, i_j \leq 10^6 \forall j = 1, 2, \dots, n$;
- Có 20% số test với $1 \leq n, i_j \leq 10 \forall j = 1, 2, \dots, n$;
- Có 24% số test với $1 \leq n, i_j \leq 1000 \forall j = 1, 2, \dots, n$;
- Có 24% số test với $1 \leq n \leq 1000$;
- Có 32% số test với ràng buộc gốc.

Bài L. WIRETAPP

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 0.7 second
Hạn chế bộ nhớ: 256 MiB

Có n tội phạm đang hoạt động trong xứ tự do. Các tội phạm được đánh số từ 1 đến n . Các tội phạm thường liên lạc qua điện thoại, nhưng chúng sẽ mã hoá đường truyền để tránh bị lộ thông tin. Mỗi tội phạm sử dụng một tần số riêng để mã hoá, gọi tần số mà tội phạm thứ i sử dụng là a_i , là một số thực chưa biết (có thể âm). Nếu biết được giá trị của a_i , có thể dễ dàng giải mã các thông tin của hắn. Công an đã thực hiện m cuộc nghe lén, mỗi cuộc là giữa hai người nào đó. Vì đã được mã hoá nên công an không nghe được, tuy nhiên dựa vào tần số truyền họ có thể xác định được tổng tần số mà hai tội phạm này đang sử dụng.

Yêu cầu: Cho biết danh sách các cuộc nghe lén, hãy giúp công an tìm ra dãy a . Nếu có nhiều dãy a , đưa ra dãy có tổng giá trị tuyệt đối của các phần tử là bé nhất có thể. Nếu vẫn có nhiều dãy a cùng có tổng giá trị tuyệt đối bé nhất, đưa ra một dãy bất kỳ trong số đó.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên n ($1 \leq n \leq 10^5$) và m ($0 \leq m \leq 2 \times 10^5$) là số lượng tội phạm và số lần nghe lén.
- m dòng tiếp theo, dòng chứa ba số nguyên x, y và w cho biết cuộc nghe lén giữa x và y thu được thông tin $a_x + a_y = w$ ($1 \leq x, y \leq n$; $-10^9 \leq w \leq 10^9$).

Kết quả

Ghi n số cách nhau bởi dấu cách là dãy a tìm được. Nếu không có dãy a nào thoả mãn, ghi "invalid". Kết quả được chấm đúng nếu sai khác của từng cuộc nghe lén là không quá 10^{-6} , đồng thời sai khác của tổng giá trị tuyệt đối của dãy so với dãy tối ưu là không quá 10^{-6} .

Ví dụ

stdin	stdout
3 1 1 2 3	1.8 1.2 0
4 6 1 2 2 1 3 2 1 4 2 2 3 2 2 4 2 3 4 2	1 1 1 1

Hạn chế

- Subtask 1: $n \leq 5$, $m \leq 14$.
- Subtask 2: $n \leq 100$.
- Subtask 3: $n \leq 1000$.
- Subtask 4: $n \leq 10000$.
- Subtask 5: Không có ràng buộc gì thêm.

Bài M. 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

<code>stdin</code>	<code>stdout</code>
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 N. 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 O. MINAND

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

Cho dãy số nguyên không âm $a = a_1, a_2, \dots, a_n$ và Q truy vấn dạng:

- $0 \ L \ R \ x$: Gán $a_i = a_i \& x$ với mọi $i = L, L+1, \dots, R$. Ở đây $\&$ là phép toán and các bit tương ứng của hai số.
- $1 \ L \ R$: Tìm số nhỏ nhất trong các số a_L, a_{L+1}, \dots, a_R .

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n, Q ;
- Dòng tiếp theo chứa n số nguyên không âm $a = a_1, a_2, \dots, a_n$;
- Mỗi dòng trong số Q dòng tiếp theo chứa một truy vấn.

Kết quả

Ghi kết quả mỗi truy vấn loại 1 trên một dòng.

Ví dụ

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

Hạn chế

- Trong tất cả các test: $n, Q \leq 10^5$; $0 \leq a_i, x \leq 10^9$.
- Có 20% số test với $n, Q \leq 5000$.
- Có 30% số test với $a_i, x \leq 100$.

Bài P. INTREE

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

Công ty Hạnh Phúc có n nhân viên, được đánh số từ 1 đến n . Nhân viên số 1 là giám đốc công ty, người này không có cấp trên. Các nhân viên khác đều có một cấp trên trực tiếp, cấp trên trực tiếp của i là p_i . Người y được gọi là thuộc phòng ban của người x nếu tồn tại một dãy $x = u_1, u_2, \dots, u_k = y$ sao cho u_i là cấp trên trực tiếp của u_{i+1} với mọi $i = 1, 2, \dots, k-1$. Dữ liệu đảm bảo không có hai người x, y nào khác nhau mà thoả mãn x thuộc phòng ban của y và y thuộc phòng ban của x . Tức là, quan hệ trong công ty tạo thành một cấu trúc cây, với gốc là đỉnh 1 và các cạnh mô tả các quan hệ cấp trên trực tiếp; mỗi phòng ban là một cây con.

Hiện tại, việc trao đổi thông tin chỉ có thể xảy ra giữa một nhân viên với cấp trên trực tiếp của anh ta. Để có thể trao đổi thông tin rộng hơn, công ty cần nâng cấp bảo mật. Đầu tiên, mỗi nhân viên được cấp một số nguyên không âm dùng để mã hoá thông tin. Số nguyên được cấp cho người thứ i là a_i . Khi hai người u và v muốn trao đổi thông tin với nhau, hệ thống sẽ thiết lập một kênh mã hoá dựa vào các số nguyên của những người cần thiết, do đó tính bảo mật của kênh này là XOR của tất cả các số nguyên của những người đó. Cụ thể, độ bảo mật của kênh truyền tin giữa hai người u và v (u và v không nhất thiết phải khác nhau) là XOR của tất cả các a_x sao cho x nằm trên đường đi đơn từ u đến v trên cây. Ở đây, phép toán XOR là phép toán có ký hiệu là \wedge trong ngôn ngữ C++.

Để dễ bảo mật thông tin nội bộ, mỗi nhân viên đều muốn biết độ bảo mật cao nhất của các kênh truyền tin bên trong phòng ban của mình. Với mỗi nhân viên x , hãy tính độ bảo mật cao nhất của các kênh truyền tin bên trong phòng ban của x ; tức là tìm đường đi đơn bên trong cây con gốc x mà có XOR của các số trên đường đi đơn đó là lớn nhất có thể và in ra giá trị XOR tìm được.

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n là số nhân viên trong công ty;
- Dòng thứ hai chứa n số nguyên không âm a_1, a_2, \dots, a_n là các số được cấp cho các nhân viên;
- Dòng thứ ba chứa $n-1$ số nguyên dương p_2, p_3, \dots, p_n là cấp trên trực tiếp của các nhân viên.

Kết quả

Ghi n dòng, dòng thứ x là độ bảo mật cao nhất của các kênh truyền tin bên trong phòng ban của x .

Ví dụ

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

Giải thích

- Phòng ban của 1: Kênh (1, 8) hoặc kênh (4, 5) hoặc kênh (8, 7).
- Phòng ban của 2: Kênh (4, 5).
- Phòng ban của 3: Kênh (3, 3) hoặc kênh (7, 7).
- Phòng ban của 4: Kênh (4, 8).
- Phòng ban của 5: Kênh (5, 5).

- Phòng ban của 6: Kênh (6, 6).
- Phòng ban của 7: Kênh (7, 7).
- Phòng ban của 8: Kênh (8, 8).

Hạn chế

- Trong tất cả các test: $n \leq 10^5$; $a_i \leq 10^9$.
- Có 12% số test với $n \leq 500$ và $a_i < 32$.
- Có 16% số test với $n \leq 5000$.
- Có 20% số test với $p_i = \lfloor \frac{i}{2} \rfloor$.
- Có 24% số test với $a_i < 32$.
- Có 28% số test với ràng buộc gốc.