

08/11:

Bài A. YENOM

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

Hệ thống tiền tệ tại một quốc gia nào đó thường đảm bảo việc lưu thông buôn bán một cách thuận tiện nhất. Theo đó nó phải có khả năng thanh toán cho tất cả các mức giá nguyên dương. Tuy vậy, do lạm phát mà mệnh giá tiền ngày càng tăng lên và các mệnh giá hiện tại có thể không giữ được tính chất cổ hữu đó nữa. Ví dụ ở một nước phát triển nọ, với các tờ tiền có mệnh giá là 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000 thì rõ ràng là không thể chỉ trả số tiền 1995

Cụ thể hơn, một số tiền x được gọi là thanh toán được bằng hệ thống tiền tệ hiện tại nếu có thể chọn số lượng cho mỗi mệnh giá sao cho tổng giá trị được chọn là x . Để đánh giá mức độ thuận tiện của một hệ thống tiền tệ trước hết người ta chọn một số nguyên dương T , thường là giới hạn các giao dịch từng được sử dụng. Sau đó họ tính số lượng các số nguyên x , $0 \leq x \leq T$, có thể thanh toán được. Hãy giúp họ thực hiện nhanh công việc khó khăn này

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n T với n là số lượng mệnh giá
- Dòng thứ hai chứa n số nguyên dương là các mệnh giá: a_1 a_2 \dots a_n

Kết quả

- Một số tự nhiên là kết quả bài toán

Ví dụ

stdin	stdout
5 10000 100 200 500 1000 2000	101

Hạn chế

- $1 \leq n, a_i \leq 2000$
- Subtask 0: $T \leq 2000$
- Subtask 1: $T \leq 20000$
- Subtask 2: $T \leq 200000$
- Subtask 3: $T \leq 10^{18}$

Bài B. WOOD2

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

Hùng có một tấm gỗ hình chữ nhật, cậu muốn cắt nó thành các miếng để sử dụng (các miếng có thể không phải hình chữ nhật). Để thuận tiện, cậu đã chia tấm gỗ thành $n \times m$ ô vuông đơn vị và tô màu cho các ô, sao cho mỗi miếng sử dụng sẽ có một màu khác nhau. Giờ đây, việc cắt gỗ đơn giản chỉ là cắt qua tất cả các đường xen giữa các màu khác nhau. Sẽ có hai công đoạn cắt: Sử dụng máy cưa lớn và sử dụng máy cưa nhỏ. Máy cưa lớn sẽ thực hiện các lát cắt đi từ biên của một miếng gỗ, cắt theo đường thẳng, và dừng lại ở bất kỳ đâu. Sau khi cắt hết các lát cắt có thể có, nếu vẫn còn những đường phải cắt thì mới sử dụng máy cưa nhỏ.

Yêu cầu: Tính số miếng gỗ rời nhau sau khi sử dụng máy cưa lớn.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n, m ($1 \leq n, m \leq 2000$);;
- Mỗi dòng trong số n dòng tiếp theo chứa một xâu gồm m ký tự (in hoa, in thường hoặc chữ số) mô tả màu của các ô.

Dữ liệu bảo đảm các ô cùng màu là liên thông cạnh.

Kết quả

Ghi một số nguyên dương là kết quả bài toán.

Ví dụ

stdin	stdout
8 7 AAAAAAA BBxxxBB BBBBBBB ccccccc cMcccaa cMcccaa ccccZZZ YYcTTT	7

Hạn chế

Có ít nhất 50% số điểm với $n, m \leq 20$.

Bài C. COMPUTE

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

Một máy tính giải phương trình được thiết kế để hoạt động theo cách sau:

- Máy nhận đầu vào là một dãy các ký tự $+$, $-$, $*$, $/$, X và các chữ số từ 0 đến 9, trong đó ký tự X xuất hiện nhiều nhất 1 lần;
- Khi gặp ký tự X hoặc chữ số, máy sẽ đẩy biểu thức (chỉ một ký tự cũng được coi là một biểu thức) này vào ngăn xếp;
- Khi gặp ký tự $+$, $-$, $*$, $/$ máy sẽ lấy hai biểu thức trong ngăn xếp ra và thực hiện phép toán tương ứng, sau đó đẩy biểu thức mới vào ngăn xếp.

Kết thúc quá trình, trong ngăn xếp sẽ còn đúng một biểu thức. Máy sẽ tìm các số hữu tỷ X sao cho biểu thức đó có giá trị bằng 0.

Dữ liệu vào

Gồm một dòng chứa dãy các ký tự đầu vào, cách nhau bởi dấu cách, độ dài không quá 30.

Kết quả

- Nếu không có giá trị X thoả mãn, in ra "none";
- Nếu có nhiều giá trị X thoả mãn, in ra "multiple";
- Nếu có đúng một giá trị X thoả mãn, in ra hai số nguyên p và q để thể hiện $X = p/q$ là kết quả (cần giảm ước để $q > 0$ và $\gcd(|p|, q) = 1$).

Ví dụ

stdin	stdout
9 3 X * 1 + 2 1 1 + * / -	35 3
2 2 1 * -	multiple
2 2 1 * *	none

Giải thích

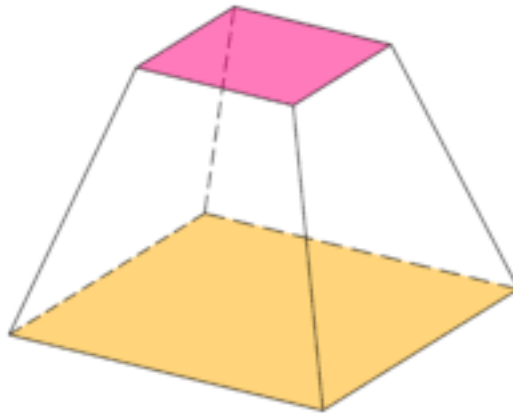
Biểu thức trong test VD là: $9 - (3x + 1)/(2 * (1 + 1))$.

Bài D. POLYWORM

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

Khu rừng Amz có thể được hiểu như một mặt phẳng tọa độ. Có n con sâu đang bò trên đó theo các quỹ đạo thẳng đều, vị trí của con thứ i tại thời điểm t là $(a_i \times t + b_i, c_i \times t + d_i)$. Để quản lý các con sâu này, người ta dùng một tấm lưới có độ đàn hồi rất cao. Khi trùm tấm lưới này lên các con sâu, nó sẽ co lại thành đa giác lồi nhỏ nhất chứa toàn bộ sâu (có thể hiểu tấm lưới như bao lồi của tập điểm này).

Việc sử dụng tấm lưới có diện tích S trong khoảng thời gian Δt sẽ mất chi phí $\Delta t \times S$. Để dễ hiểu, có thể tưởng tượng rằng cả tấm lưới di chuyển theo trục vuông góc với nó (trục thời gian Oz), với vận tốc bằng 1. Lúc này, các con sâu di chuyển khiến hình dạng tấm lưới thay đổi theo thời gian, đồng thời vị trí của tấm lưới cũng di chuyển về nên một hình khối 3 chiều. Chi phí sẽ tính bằng thể tích của hình này.



Dữ liệu vào

- Dòng đầu ghi hai số nguyên dương: n T
- n dòng tiếp theo, dòng thứ i chứa bốn số nguyên: a_i b_i c_i d_i

Kết quả

- Ghi chi phí sử dụng lưới theo định dạng dấu chấm thập phân
- Đáp án A được chấp nhận với kết quả B nếu $\frac{|A-B|}{\max(A,B)} \leq 10^{-6}$

Ví dụ

stdin	stdout
4 5 0 0 0 0 1 1 1 1 0 2 2 0 2 2 2 2	76.000

Hạn chế

- $1 \leq n \leq 50$
- $-1000 \leq a_i, b_i, c_i, d_i \leq 1000$
- $1 \leq T \leq 10000$

Bài E. YENOM2

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

Hệ thống tiền tệ tại một quốc gia nào đó thường đảm bảo việc lưu thông buôn bán một cách thuận tiện nhất. Theo đó nó phải có khả năng thanh toán cho tất cả các mức giá nguyên dương. Tuy vậy, do lạm phát mà mệnh giá tiền ngày càng tăng lên và các mệnh giá hiện tại có thể không giữ được tính chất cổ hữu đó nữa. Ví dụ ở một nước phát triển nọ, với các tờ tiền có mệnh giá là 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000 thì rõ ràng là không thể chi trả số tiền 1995

Cụ thể hơn, một số tiền x được gọi là thanh toán được bằng hệ thống tiền tệ hiện tại nếu bên trả tiền có thể chọn số lượng tờ cho mỗi mệnh giá để tổng giá trị được chọn là a , bên nhận tiền có thể chọn số lượng tờ cho mỗi mệnh giá để tổng giá trị được chọn là b , sao cho $a - b = x$. Để đánh giá mức độ thuận tiện của một hệ thống tiền tệ trước hết người ta chọn một số nguyên dương T , thường là giới hạn các giao dịch từng được sử dụng. Sau đó họ tính số lượng các số nguyên x , $0 \leq x \leq T$, có thể thanh toán được. Hãy giúp họ thực hiện nhanh công việc khó khăn này

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n T với n là số lượng mệnh giá
- Dòng thứ hai chứa n số nguyên dương là các mệnh giá: a_1 a_2 \dots a_n

Kết quả

- Một số tự nhiên là kết quả bài toán

Ví dụ

stdin	stdout
2 10 4 6	6
5 10000 100 200 500 1000 2000	101

Giải thích

Các số tiền có thể thanh toán được trong VD1 là: 0, 2, 4, 6, 8, 10

Hạn chế

- $1 \leq n, a_i \leq 2000$
- Subtask 0: $T \leq 2000$
- Subtask 1: $T \leq 20000$
- Subtask 2: $T \leq 200000$
- Subtask 3: $T \leq 10^{18}$

Bài F. STRX

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

Với s là một chuỗi ký tự và k là một số tự nhiên nhỏ hơn độ dài của s , ký hiệu $\text{shift}^k(s)$ là chuỗi thu được bằng cách chuyển k ký tự đầu tiên của s xuống cuối s . Ví dụ, $\text{shift}^3(abxybcffe) = ybcffeabx$.

Cho t và s là hai chuỗi nhị phân cùng độ dài. Tìm hai số i và j sao cho $0 \leq i \leq j < |s|$ và $\text{shift}^i(s) | \text{shift}^j(s) = t$. Ở đây $|$ là phép toán OR từng bit tương ứng.

Dữ liệu vào

- Dòng đầu chứa chuỗi t ;
- Dòng thứ hai chứa chuỗi s .

Kết quả

Nếu không tồn tại i và j , in ra -1. Ngược lại, in ra i và j trên một dòng. Nếu có nhiều đáp án, in ra một đáp án tùy ý.

Ví dụ

stdin	stdout
111011 100101	1 5

Hạn chế

- 50% test với $|S| \leq 500$;
- 50% test với $|S| \leq 5000$;

Bài G. DELGAME

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

Cho đồ thị vô hướng $G = (V, E)$ và một số nguyên dương k . Có hai người chơi 1 trò chơi trên G như sau:

- Hai người luân phiên thực hiện lượt chơi
- Đến lượt mình, người chơi chọn một số cạnh của G và xóa đi, sao cho:
 - Số cạnh bị xóa là số nguyên dương không vượt quá k
 - Các cạnh đó thuộc cùng 1 thành phần liên thông trên G
 - Sau khi xóa, số thành phần liên thông của G vẫn không đổi
- Ai không thực hiện được nước đi nữa thì thua

Biết cả hai đều chơi tối ưu, hãy cho biết kết quả của trò chơi

Dữ liệu vào

- Gồm 4 bộ dữ liệu, mỗi bộ bao gồm:
 - Dòng đầu: $n \ m \ k$ với n, m là số đỉnh và số cạnh của G
 - m dòng tiếp theo ghi các cạnh của G : $u \ v$

Kết quả

Gồm 4 dòng, ghi 1/0 tương ứng là người đi trước thắng/thua

Ví dụ

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

Hạn chế

- $1 \leq n, m, k \leq 10^5$
- 40% số test có G liên thông
- 30% số test có $k = n$

Bài H. DCATCHER

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

Dream catcher là vật trang trí phổ biến và hợp xu thế của giới trẻ ngày nay. Nó bao gồm một dây treo nối với một vòng tròn; trên vòng tròn này sẽ treo những chiếc lông vũ hoặc là những Dream catcher khác nhỏ hơn (một lông vũ cũng được xem là một Dream catcher). Để sản xuất Dream catcher hàng loạt, người ta đã chế tạo một cỗ máy hoạt động như sau:

- Có rất nhiều vòng, dây treo và lông vũ cung cấp làm nguyên liệu cho máy một cách thoải mái;
- Máy nhận lệnh sản xuất là một xâu chỉ chứa ký tự 'a' và các chữ số từ 1 đến 9. Khi nhận lệnh, máy sẽ đọc từng ký tự từ trái sang phải, và:
 - Gặp ký tự 'a', nó sẽ lấy một cái lông vũ từ nguồn nguyên liệu, bỏ vào ngăn xếp của nó;
 - Gặp ký tự x là số, nó sẽ lấy một cái vòng và dây treo từ nguồn nguyên liệu, sau đó lấy ra x đối tượng từ ngăn xếp và treo vào vòng này, sau đó nó bỏ Dream catcher thu được vào trong ngăn xếp của nó.
- Kết thúc quá trình, trong ngăn xếp sẽ còn đúng một Dream catcher (nếu không, lệnh sản xuất được coi là không hợp lệ).

Hai Dream catcher A và B được coi là đẳng cấu nếu cả hai đều là lông vũ; hoặc vòng của A và của B có số Dream catcher con treo vào bằng nhau, và tồn tại một cách xoay vòng để các Dream catcher con này tương ứng đẳng cấu với nhau. Rõ ràng là mỗi Dream catcher có thể có nhiều lệnh sản xuất khác nhau. Để tối ưu chi phí vận hành, hãy thay thế lệnh sản xuất được cho thành một lệnh sản xuất mới sao cho kích thước ngăn xếp của máy là nhỏ nhất có thể mà vẫn bảo đảm sản xuất được.

Dữ liệu vào

Gồm một dòng duy nhất chứa một xâu là lệnh sản xuất hợp lệ.

Kết quả

- Dòng đầu chứa kích thước của ngăn xếp;
- Dòng tiếp theo chứa lệnh sản xuất mới.

Ví dụ

<code>stdin</code>	<code>stdout</code>
aa2aaa3aaa33	4 aaa3aaa3aa23

Hạn chế

- 50% số test với độ dài xâu không quá 100;
- 50% số test khác với độ dài xâu không quá 100000;

09/11:

Bài I. ZFU

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

Hệ thống bảo mật của quân đội XYZ gồm một dãy n khóa (gọi là dãy S), các khóa được đánh số từ 1 đến $|S|$. Có thể có nhiều loại khóa khác nhau, mỗi loại đặc trưng bởi một ký tự khác nhau. Để phá hủy hệ thống bảo mật đó, quân đội nhân dân ABC đã chế tạo một robot. Robot này sẽ phá lần lượt từng loại khóa theo thứ tự đã được lập trình sẵn, và nếu loại khóa cần phá hủy khác với loại phá khóa đang được sử dụng theo lệnh lập trình, robot sẽ không thể phá được khóa đó và dừng toàn bộ lệnh sau đó.

Ban đầu, robot được lập trình theo đúng dãy S . Dĩ nhiên nó sẽ phá được hết các khóa nếu được đặt vào vị trí đầu tiên của S , tuy nhiên việc tiếp cận dãy khóa này không hề dễ dàng. Theo đó, chim ruồi (một loại máy bay không người lái) sẽ mang robot và thả xuống khu vực có dãy khóa. Nếu bay đến vị trí i trên dãy khóa, chim ruồi có thể thả robot xuống các vị trí $j \geq i$, tức các vị trí từ i trở về sau của dãy. Tất nhiên là i càng nhỏ sẽ càng có cơ may phá được nhiều khóa hơn, tuy nhiên mức độ nguy hiểm cũng cao hơn. Để cân bằng giữa rủi ro và kết quả, quân đội nhân dân ABC muốn trả lời Q câu hỏi, mỗi câu hỏi có dạng: "Nếu chim ruồi bay đến vị trí i thì có nhiều nhất bao nhiêu khóa sẽ bị phá".

Dữ liệu vào

- Dòng đầu tiên chứa một xâu ký tự mô tả cho dãy khóa S
- Dòng tiếp theo chứa số nguyên dương Q
- Q dòng tiếp theo, mỗi dòng chứa một số i

Kết quả

- Ghi ra Q dòng, mỗi dòng ghi kết quả cho một câu hỏi tương ứng

Ví dụ

stdin	stdout
abaabababab	10
3	5
1	2
2	
8	

Hạn chế

- $1 \leq |S|, Q \leq 10^6$
- Có 30% số test với $1 \leq |S|, Q \leq 500$
- Có 30% số test với $501 \leq |S|, Q \leq 5000$

Bài J. PRESUFF

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

Cho một tập A các chuỗi ký tự. Ta nói chuỗi p khác rỗng là một tiền tố của tập A nếu có thể thêm một số ký tự vào sau p để trở thành một chuỗi thuộc A . Ta nói chuỗi s khác rỗng là một hậu tố của tập A nếu có thể thêm một số ký tự vào trước s để trở thành một chuỗi thuộc A . Lưu ý, chuỗi rỗng không được xem là một tiền tố/hậu tố của một chuỗi thuộc A thì có. Tiến hành xây dựng tập B bằng cách nối các chuỗi p với f (viết liền nhau) với mọi p là tiền tố và f là hậu tố của A . Tức là, mỗi chuỗi của B thu được bằng cách viết một tiền tố và một hậu tố của A liền nhau. Hãy đếm xem có bao nhiêu chuỗi phân biệt trong cả hai tập A và B .

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n là số chuỗi trong tập A ($n \leq 10000$);
- Mỗi dòng trong số n dòng tiếp theo chứa một chuỗi trong tập A , độ dài chuỗi không quá 40.

Kết quả

Ghi một số nguyên duy nhất là số chuỗi phân biệt trong tập $A \cup B$.

Ví dụ

stdin	stdout
4 ab aba abcde c	71

Hạn chế

Có ít nhất 50% số test với $n \leq 500$ và độ dài các chuỗi không quá 10.

Bài K. GCD2

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

Có một số nguyên dương x bị ẩn đi, thông tin duy nhất là số này nằm trong phạm vi $[1; n]$. Để đoán ra x , bạn được quyền hỏi các câu hỏi, mỗi câu hỏi là một số nguyên dương y nằm trong phạm vi $[1; n]$, máy sẽ trả lời $\gcd(x, y)$. Gọi $f(n)$ là số lần hỏi ít nhất để chắc chắn tìm ra được số x trong phạm vi $[1, n]$ trong trường hợp xấu nhất. Số lần hỏi của bạn phải không quá $f(n)$.

Tương tác:

- Đầu tiên bạn đọc vào số nguyên dương n ($n \leq 10000$) từ đầu vào chuẩn;
- Sau đó bạn in ra $f(n)$, máy sẽ kiểm tra, nếu bạn tính sai máy sẽ dừng tương tác;
- Mỗi lần hỏi, bạn cần in ra theo định dạng: ? y sau đó đọc câu trả lời từ đầu vào chuẩn;
- Để trả lời, bạn cần in ra theo định dạng: ! x sau đó kết thúc chương trình.

Ví dụ

stdin	stdout
6	2
2	? 6
2	? 4
	! 2

Hạn chế

Nếu chỉ tính đúng $f(n)$ nhưng không trả lời được x , bạn sẽ nhận được 50% số điểm của test đó.

Bài L. SPGM

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

Hai người S và P cùng nhau chơi một trò chơi và nhờ GM làm trọng tài. Trò chơi bắt đầu bởi một số nguyên dương N là số được thống nhất bởi ba người họ. Trọng tài sẽ chọn ra hai số nguyên dương KHÁC nhau trong khoảng từ 1 đến N , sau đó nói cho S biết tổng và P biết tích của hai số đó. Hai người S và P sẽ luân phiên nhau cố gắng đoán ra hai số này, S đoán trước. Ví dụ với $N = 10$, GM chọn 3 và 6 thì trò chơi sẽ diễn ra như sau:

- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao biết rồi, đó là 3 và 6"

Hoặc với $N = 15$, GM chọn 3 và 5 thì trò chơi sẽ diễn ra như sau:

- S: "Tao không biết"
- P: "Tao không biết"
- S: "Tao không biết"
- P: "Thôi dẹp đi, có đoán đến sáng mai cũng không ra đâu"

Cho biết N là số mà cả ba người đều biết, x và y là hai số mà GM đã chọn (tức là GM sẽ nói cho S biết giá trị của $x + y$ và nói cho P biết giá trị của $x \times y$), hãy đếm số lần mà câu "Tao không biết" được nói ra. Biết rằng cả hai người chơi đều rất thông minh và trung thực

Dữ liệu vào

- Dòng đầu chứa số lượng test: T
- T dòng tiếp theo mỗi dòng chứa: $N \ x \ y$

Kết quả

- Gồm T dòng ghi số lần mà câu "Tao không biết" được nói ra tương ứng cho T test

Ví dụ

stdin	stdout
2	4
10 3 6	3
15 3 5	

Hạn chế

- $2 \leq N \leq 200, 1 \leq T \leq 10$
- Có ít nhất 60% số test mà S hoặc P sẽ đoán ra được x, y

Bài M. BIRD

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

Có N con chim đậu thành một hàng. Con thứ i có sức mạnh là a_i . Chúng rất đoàn kết, vì vậy khi bạn bắn con chim thứ i , các con chim $i-1$ (nếu $i > 1$), i , và $i+1$ (nếu $i < n$) sẽ hợp sức lại chống trả. Vì vậy bạn mất $a_{i-1} + a_i + a_{i+1}$ năng lượng để bắn chết con chim thứ i . Sau đó, các con chim lại đứng xích lại gần nhau thành 1 hàng liên tiếp.

Thai9cdb luôn bắn con chim có sức mạnh lớn nhất, nếu có nhiều con cùng có sức mạnh lớn nhất, anh ta sẽ bắn con có số thứ tự nhỏ nhất. Cho đến khi lũ chim chết hết anh ta sẽ giàu to. Tuy nhiên anh không mang theo máy tính nên không biết mình sẽ mất bao nhiêu năng lượng. Hãy giúp anh ta!

Dữ liệu vào

- Dòng đầu chứa: n
- Dòng tiếp theo chứa dãy a

Kết quả

- Một số nguyên duy nhất là tổng năng lượng cần dùng

Ví dụ

stdin	stdout
5 1 2 3 4 5	25

Hạn chế

- $1 \leq N \leq 10^6$, $1 \leq a_i \leq 10^9$
- 25% test có $N \leq 1000$

Bài N. DOUBLEGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 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. Đồng xu sau khi được di chuyển sẽ tự động nhân đôi lên. Có nghĩa là nếu bạn chuyển một đồng xu từ x sang y , a_x sẽ giảm một còn a_y sẽ tăng hai
- 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: $u \ v$
- 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 3 1	YES

Hạn chế

- $1 \leq n, m, a_i \leq 10^5$

Bài O. NXTBINARY

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

Gọi $S(n)$ là tập các xâu nhị phân độ dài n , không có hai bit 1 nào kề nhau. Cho x là một xâu trong tập $S(n)$, tìm xâu tiếp theo thứ k kể từ xâu x , tính theo thứ tự từ điển trong $S(n)$. Tức là sắp xếp lại các xâu trong $S(n)$ và tìm xâu đứng ở sau k vị trí so với x .

Dữ liệu vào

- Dòng đầu chứa ba số nguyên không âm n, a, b ($n, a, b \leq 10^5$). Khi đó $k = f_a + b$ với f là dãy fibonacci định nghĩa bởi: $f_0 = f_1 = 1, f_n = f_{n-1} + f_{n-2}$;
- Dòng thứ hai chứa xâu x .

Kết quả

Ghi một xâu duy nhất là kết quả tìm được, hoặc -1 nếu vị trí thứ k sau x vượt quá số lượng phần tử của $S(n)$.

Ví dụ

stdin	stdout
6 1 1 001010	010001

Hạn chế

- Subtask 1: $n \leq 23$;
- Subtask 2: $a = 0$;
- Subtask 3: $b = 0$;
- Subtask 4: Ràng buộc gốc.

Bài P. 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.

10/11:

Bài Q. MGCD

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

Cho dãy A gồm n số nguyên dương: a_1, a_2, \dots, a_n . Xét tất cả các dãy con khác rỗng của A , với mỗi dãy ta sẽ tìm ước chung lớn nhất của tất cả các phần tử. Cần tính tích của tất cả các ước chung đó. Cụ thể hơn, hãy tính:

$$\prod_{S \subseteq A, S \neq \emptyset} \gcd(S)$$

trong đó $\gcd(S)$ là số nguyên dương lớn nhất là ước của tất cả các số trong S

Dữ liệu vào

- Dòng đầu tiên chứa n
- Dòng tiếp theo chứa n số nguyên dương là dãy A

Kết quả

- Một số nguyên là kết quả bài toán sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

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

Hạn chế

- $1 \leq n \leq 10^5$ $1 \leq a_i \leq 10^5$
- Subtask 1: a_i đôi một phân biệt

Bài R. WDIC

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 256MB

Với x là một từ (một chuỗi ký tự chỉ chứa các chữ cái in hoa), ta có thể xoá đi một số (có thể là 0) ký tự của x và hoán vị các ký tự còn lại để sinh ra các biến thể. Độ phổ biến của x là số biến thể khác nhau của x nằm trong từ điển. Tuy nhiên từ điển đã bị thất lạc, bạn chỉ biết nó là một tập gồm n từ phân biệt. Người ta đã tìm thấy một tài liệu còn sót lại có chứa m từ thông dụng. Bạn cần khôi phục lại từ điển để tổng độ phổ biến của m từ đó là lớn nhất có thể. Do m từ thông dụng đã có trong tài liệu, nó sẽ không được xuất hiện trong từ điển nữa.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n và m ($n \leq 100, m \leq 1000$);
- Mỗi dòng trong số m dòng tiếp theo chứa một từ thông dụng, độ dài không quá 100.

Kết quả

Gồm n dòng ghi n từ trong từ điển. Nếu có nhiều từ điển thoả mãn, chỉ cần in ra một trong số đó.

Ví dụ

stdin	stdout
5 4	E
TOI	I
YEU	IT
VIET	T
NAM	TI

Hạn chế

Có ít nhất 50% số test với $n \leq 10, m \leq 100$.

Bài S. PERMUTATION1

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

Có một hoán vị $P = (p_1, p_2, \dots, p_n)$ của $(1, 2, \dots, n)$ bị ẩn đi, thông tin duy nhất là độ dài n . Để đoán ra P , bạn được quyền hỏi các câu hỏi, mỗi câu hỏi là ba số nguyên (i, j, k) thoả mãn $1 \leq i < j < k \leq n$, máy sẽ trả lời trung vị của (p_i, p_j, p_k) là thuộc về chỉ số i hay j hay k . Nhắc lại, trung vị của một dãy có m phần tử là phần tử lớn thứ $(m + 1)/2$ của dãy.

Dễ thấy dù hỏi hết tất cả các bộ ba, thì hoán vị P vẫn không duy nhất mà sẽ là một họ các hoán vị. Chẳng hạn, nếu P thoả mãn tất cả các câu hỏi thì viết P theo thứ tự ngược lại cũng thoả mãn. Do đó, bạn chỉ cần trả lời P bất kỳ trong họ đó.

Tương tác:

- Đầu tiên bạn đọc vào số nguyên dương n ($n \leq 200$) từ đầu vào chuẩn;
- Để trả lời, bạn in ra "OK " sau đó là hoán vị P ;
- Để hỏi, bạn in ra ba số i, j, k sau đó đọc kết quả từ đầu vào chuẩn;
- Bạn được hỏi không quá 2000 lần.

Lưu ý, sau mỗi lần in ra bạn cần flush(stdout) hoặc cout<<endl để đẩy dữ liệu sang cho trình chấm.

Ví dụ

stdin	stdout
5	1 2 3
2	2 3 4
3	3 4 5
4	OK 5 4 3 2 1

Bài T. EULER2

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

Quang muốn chia N cái kẹo thành k phần để đi chúc tết họ hàng. Cậu thắc mắc có bao nhiêu cách chia? Thứ tự của các phần là không quan trọng, tức là hai cách chia là hoán vị của nhau được tính là một.

Dữ liệu vào

- Chứa hai số tự nhiên: N k

Kết quả

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

Ví dụ

stdin	stdout
5 2	3

Hạn chế

- $1 \leq N \leq 10^{18}$, $1 \leq k \leq 15$
- Subtask 0: $N \leq 10^6$
- Subtask 1: $k \leq 10$

Bài U. SEQMX

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

Cho dãy số nguyên $a: a_1, a_2, \dots, a_n$ và một số nguyên dương k . Hãy tìm một đoạn con của a , có ít nhất k phần tử và có trung bình cộng lớn nhất có thể.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n k
- Dòng tiếp theo chứa dãy $a: a_1 a_2 \dots a_n$

Kết quả

Ghi trung bình cộng lớn nhất tìm được, quy tròn đến số thập phân thứ năm

Ví dụ

stdin	stdout
7 3 2 4 5 1 3 4 1	3.66667

Hạn chế

- $1 \leq k \leq n \leq 10^5, -10^5 \leq a_i \leq 10^5$
- Có 50% số test với $n \leq 5000$

Bài V. 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

test	answer
3 10 0 5 1 2 1 3	2
1 1 -1 10	20

Hạn chế

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

Bài W. TRIPLEGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 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. Đồng xu sau khi được di chuyển sẽ tự động nhân ba lên. Có nghĩa là nếu bạn chuyển một đồng xu từ x sang y , a_x sẽ giảm một còn a_y sẽ tăng ba
- 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: $u \ v$
- 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 3 1	YES

Hạn chế

- $1 \leq n, m, a_i \leq 10^5$

Bài X. FDIV

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

Cho số X có dạng $X = \frac{p_1!p_2!\dots p_n!}{q_1!q_2!\dots q_m!}$. Cần tìm dãy $r_1, s_1, r_2, s_2, \dots, r_k, s_k$ có thứ tự từ điển lớn nhất sao cho $Y = r_1!^{s_1} r_2!^{s_2} \dots r_k!^{s_k}$ là ước của X (với $r_i > 1, s_i > 0$). Trường hợp dãy rỗng ($k = 0$) ứng với $Y = 1$.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương là n và m ($n, m \leq 1000$);
- Dòng tiếp theo chứa n số nguyên dương là dãy p ($p_i \leq 10000$);
- Dòng tiếp theo chứa m số nguyên dương là dãy q ($q_i \leq 10000$).

Kết quả

Ghi -1 nếu X không phải là số nguyên. Ngược lại:

- Dòng đầu tiên chứa số k ;
- Dòng thứ i trong số k dòng tiếp theo chứa r_i s_i .

Ví dụ

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

Hạn chế

- Có 25% số test với $n, m \leq 10$ và $p_i, q_i \leq 20$;