

ĐỀ THI ĐỀ XUẤT

TỔNG QUAN VỀ BÀI THI

Tên bài	Tệp chương trình	Tệp dữ liệu vào	Tệp dữ liệu ra	Điểm
Đoạn bội	MULTIPLY.*	MULTIPLY.INP	MULTIPLY.OUT	6
Cây táo	APPLE.*	APPLE.INP	APPLE.OUT	7
Di cư	DICU.*	DICU.INP	DICU.OUT	7

Phần mở rộng của tệp chương trình được đặt tùy theo ngôn ngữ lập trình được sử dụng.

Bài 1: Đoạn bội (6đ)

Cho hai đoạn chứa số nguyên dương $[a, a + 1, \dots, b]$ và $[c, c + 1, \dots, d]$. Hãy xác định xem tích $c.(c + 1) \dots d$ có chia hết cho tích $a.(a + 1) \dots b$ hay không.

Input: tệp **Multiply.inp**

- Dòng đầu chứa số nguyên t ($1 \leq t \leq 10$) là số lượng test cases.
- Mỗi dòng trong t dòng tiếp theo gồm bốn số nguyên dương a_i, b_i, c_i, d_i .

Output: tệp **Multiply.out**

Gồm t dòng. Với test cases thứ i , in ra Y nếu tích $c.(c + 1) \dots d$ có chia hết cho tích $a.(a + 1) \dots b$, ngược lại in N .

Ví dụ

Multiply.inp	Multiply.out
2	Y
9 10 3 6	N
2 5 7 9	
6	Y
1 2 3 4	N
1 4 2 3	Y
2 3 1 4	Y
1 3 2 4	Y
19 22 55 57	Y
55 57 19 22	

Giải thích ví dụ 1:

Ta có $9.10 = 90$ và $3.4.5.6 = 360$. Kết quả là Y bởi vì 360 chia hết cho 90 . Ta tính được $2.3.4.5 = 120$ không thể bị chia hết bởi $7.8.9 = 504$. Do đó kết quả là N .

Ràng buộc:

- $1 \leq a_i \leq b_i \leq 10^7$
- $1 \leq c_i \leq d_i \leq 10^7$
- Time: 1.5s

Bài 2: Cây táo (7đ)

Nam vừa mua được một cây rìu nên hôm nay anh ấy quyết định thử nó trên cây táo của mẹ mình. Cây táo gồm có n trái táo được đánh số từ $1 \rightarrow n$. Những trái táo liên kết với nhau qua các cành tạo thành **đồ thị dạng cây**.

Khi cắt một cành nối giữa hai trái táo sẽ tạo thành hai đùm táo và mỗi đùm có một quả táo với khối lượng lớn nhất, Nam sẽ tốn khoảng thời gian bằng **tổng khối lượng hai quả táo đó**.

Do sợ mẹ phát hiện nên anh ta cần bạn giúp tìm **thời gian ngắn nhất** để cắt hết $n-1$ cành.

Input: tệp Apple.inp

- Dòng đầu gồm một số nguyên n là số trái táo. Các trái táo được đánh số từ $1 \rightarrow n$.
- Dòng thứ hai gồm n số nguyên m_i là khối lượng của quả táo thứ i ($1 \leq m_i \leq 10^9$).
- Mỗi dòng trong $n - 1$ dòng còn lại gồm hai số nguyên x và y cho biết có cành nối trực tiếp giữa hai quả táo x và y ($1 \leq x, y \leq n$).

Output: Apple.out

- In ra thời gian ngắn nhất để cắt hết $n - 1$ cành.

Ví dụ:

Apple.inp	Apple.out
3 1 2 3 1 2 2 3	8
4 2 2 3 2 1 3 3 2 4 3	15
5 5 2 3 1 4 2 1 3 1 2 4 2 5	26

Giải thích ví dụ 1

- Có 2 cách để cắt. Đầu tiên anh ta cắt cành (1,2), tốn $1 + 3 = 4$ giây, sau đó cắt cành (2,3) tốn $2 + 3 = 5$ giây. Vậy tổng cộng là 9 giây trong trường hợp này.
- Ngoài ra, anh ta có thể cắt (2,3), sau đó cắt (1,2). Tổng cộng thời gian trong trường hợp này là $(2 + 3) + (1 + 2) = 8$.

Subtask

- Subtask 1: (15% số điểm) $1 \leq n \leq 10$.
- Subtask 2: (15% số điểm) Hai quả táo i và $i + 1$ ($1 \leq i \leq n - 1$) được nối trực tiếp với nhau.
- Subtask 3: (30% số điểm) $1 \leq n \leq 1000$.
- Subtask 4: (100% số điểm) $1 \leq n \leq 100000$.

Bài 3: Di cư (7đ)

Bạn được trao bản đồ di cư của loài vịt và có nhiệm vụ giúp những chú vịt ấy di cư sang hòn đảo mới. Hòn đảo nơi loài vịt đang sống được đánh dấu bởi ký tự 'o'. Vịt có thể bắt đầu chuyển hành trình của mình theo bốn hướng. Hướng bắc được biểu hiện bởi ký tự '^', hướng đông '>', hướng nam 'v', hướng tây '<'. Khi chú vịt nằm ở ô có ký tự nào thì sẽ di chuyển một ô theo hướng mà ký tự đó đã định. Dòng nước xoáy được đánh dấu bởi ký tự '.'. Nếu chú vịt ở tại ô có dòng nước xoáy, ra ngoài bản đồ hoặc quay về hòn đảo ban đầu thì chú sẽ dừng chuyển di cư của mình. Hòn đảo mà chú vịt muốn đến được đánh dấu bằng ký tự 'x'.

Nhiệm vụ của bạn bây giờ là hãy **thay đổi ít nhất các ký tự** trên bản đồ để chú vịt có thể **đi đến hòn đảo của chú vịt mong muốn** nhé.

Tuy nhiên, những ô với ký tự 'o' và 'x' không thể bị sửa đổi. Tất cả ô với ký tự '<>v^.' có thể được thay thế bằng các ký tự trong cùng tập '<>v^.'.

Input: tệp Dicu.inp

- Dòng đầu gồm hai số nguyên r và s ($3 \leq r, s \leq 10^7$) là số dòng và số cột trên bản đồ.
- Mỗi dòng trong r dòng tiếp theo chứa s kí tự thuộc tập ' $\mathbf{o} < > \mathbf{v} \mathbf{^} \mathbf{x}$ ' biểu diễn hướng đi trên bản đồ. Có chính xác một kí tự ' $\mathbf{o}$ ' và một kí tự ' $\mathbf{x}$ ' trên bản đồ và chúng không nằm cạnh nhau.

Output: tệp **Dicu.out**

- In ra một số nguyên duy nhất là số lần thay đổi ít nhất để chú vịt có thể di chuyển từ hòn đảo đang sống đến hòn đảo mong muốn.

Ví dụ:

Dicu.inp	Dicu.out	Giải thích
3 3 >vo vv> x>>	1	Bản đồ sau khi sửa đổi:>vo vv> x<>Ban đầu chú vịt sẽ đi sang hướng tây đến ô (1,2). Tại ô này có hướng sang phía nam nên chú vịt đến ô (2,2). Chú vịt tiếp tục đi theo các ô đã định rồi kết thúc tại ô có kí tự ' $\mathbf{x}$ '.
3 6 >>vv<< ^ovvx^ ^<<>>^	2	Bản đồ sau khi sửa đổi:>>vv<< ^o>>x^ ^<<>>^
4 4 x.v. .>.< >.<. .^o	4	Bản đồ sau khi sửa đổi:x<<. .>^< >.<^ .^o

Ràng buộc:

- Time: 2s
- Subtask 1: (15% số điểm) $3 \leq r, s \leq 20$.
- Subtask 2: (50% số điểm) $3 \leq r, s \leq 2000$.
- Subtask 3: (100% số điểm) $3 \leq r, s$ và $r * s \leq 10^7$.

----- Hết -----

Ghi chú: Cán bộ coi thi không giải thích gì thêm