

BÀI TẬP LẬP TRÌNH

DÂY DẪN

Cho n đoạn dây điện ($1 \leq n \leq 10^5$). Đoạn thứ i có độ dài $l_i \text{ cm}$ ($0 < l_i < 10^9$). Cần phải cắt các đoạn đã cho thành các đoạn sao cho có được k đoạn dây bằng nhau. Có thể không cần cắt hết các đoạn dây đã cho. Mỗi đoạn dây bị cắt có thể có phần còn thừa khác 0.

Yêu cầu: Xác định độ dài lớn nhất của đoạn dây có thể nhận được. Nếu không có cách cắt thì đưa ra số 0.

Dữ liệu: Vào từ tập tin văn bản **WIRES.INP** gồm nhiều tests, mỗi test trên một nhóm dòng

- Dòng đầu tiên chứa 2 số nguyên n và k
- Dòng thứ i trong n dòng tiếp theo chứa số nguyên l_i

Kết quả: Ghi ra tập tin văn bản **WIRES.OUT**, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên

Ví dụ:

WIRES . INP	WIRES . OUT
4 11 802 743 547 539	200

DÃY SỐ HAMMING

Dãy số nguyên dương tăng dần, trong đó ước nguyên tố của mỗi số không quá 5 được gọi là dãy Hamming. Như vậy, $10 = 2 \times 5$ sẽ là một số trong dãy Hamming, còn $26 = 2 \times 13$ – không thuộc dãy Hamming.

Phần đầu của dãy Hamming là 1,2,3,4,5,6,8,9,10,12,15, ...

Yêu cầu: Cho số nguyên $x (1 \leq x \leq 10^{18})$. Hãy xác định số thứ tự của x trong dãy Hamming.

Dữ liệu: Vào từ tập tin văn bản **HAMMING.INP**:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 10^5$).
- Mỗi dòng tiếp theo chứa một số nguyên x .

Kết quả: Ghi ra tập tin văn bản **HAMMING.OUT**, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên hoặc thông báo Not in sequence.

Ví dụ:

HAMMING . INP	HAMMING . OUT
11	1
1	2
2	6
6	Not in sequence
7	7
8	8
9	9
10	Not in sequence
11	10
12	Not in sequence
13	Not in sequence
14	

MARIO

Mario đã từng là một trò chơi rất phổ biến của học sinh. Trong một màn chơi, Mario phải vượt qua lần lượt n ống khói nằm liên tiếp nhau trên hành trình về đích. Các ống khói được đánh số từ 1 đến n , ống khói thứ i có chiều cao h_i so với mặt đất, giữa các ống khói là vực thẳm.

Ban đầu, Mario có K đơn vị năng lượng và đứng trên mặt đất. Mario phải nhảy lên ống khói số 1 để bắt đầu hành trình. Nếu Mario đang đứng trên ống khói thứ i thì tiếp theo phải nhảy đến ống khói thứ $i + 1$. Giả sử Mario đang có M đơn vị năng lượng, sau khi nhảy đến ống khói có chiều cao h thì mức năng lượng của Mario sẽ được tăng hoặc giảm đi theo quy luật sau:

- Nếu $h \leq M$ thì mức năng lượng của Mario được tăng thêm $|M - h|$ đơn vị.
- Ngược lại nếu $h > M$ thì Mario sẽ bị mất đi $|M - h|$ đơn vị năng lượng.

Như vậy, trong quá trình nhảy, nếu mức năng lượng hiện tại của Mario nhỏ hơn lượng bị mất đi thì Mario không thể thực hiện bước nhảy đến ống khói tiếp theo, nếu nhảy sẽ bị rơi xuống vực thẳm và mất mạng.

Yêu cầu: Hãy tìm K nhỏ nhất để Mario có thể nhảy được đến ống khói thứ n .

Dữ liệu: Vào từ tập tin văn bản **MARIO.INP**

- Dòng đầu chứa số nguyên $n (1 \leq n \leq 10^6)$
- Dòng tiếp theo chứa dãy gồm n số nguyên $h_1, h_2, \dots, h_n (1 \leq h_i \leq 10^6)$

Kết quả: Ghi ra tập tin văn bản **MARIO.OUT** giá trị K nhỏ nhất thỏa yêu cầu.

Ví dụ:

MARIO . INP	MARIO . OUT
5 4 2 8 6 3	4

BẢNG SỐ

Cho một bảng kích thước vô hạn được chia làm lưới ô vuông đơn vị. Các hàng của bảng được đánh số từ 1 từ trên xuống và các cột của bảng được đánh số từ 1 từ trái qua phải. Ô nằm trên giao điểm của hàng i , và cột j được gọi là ô (i, j) . Người ta điền các số nguyên liên tiếp bắt đầu từ 1 vào bảng theo quy luật sau:

1	3	6	10	15	...
2	5	9	14		...
4	8	13			...
7	12				...
11					...
⋮	⋮	⋮	⋮	⋮	⋮

Yêu cầu 1: Xác định giá trị ghi trên ô (x, y) của bảng

Yêu cầu 2: Xác định hàng và cột của ô chứa giá trị z .

Dữ liệu: Vào từ file văn bản **MAPPING.INP**

- Dòng 1 chứa hai số nguyên dương $x, y \leq 10^9$
- Dòng 2 chứa số nguyên dương $z \leq 10^9$

Kết quả: Ghi ra file văn bản **MAPPING.OUT**

- Dòng 1 ghi giá trị trên ô (x, y)
- Dòng 2 ghi chỉ số hàng và chỉ số cột của ô chứa giá trị z

Ví dụ:

MAPPING . INP	MAPPING . OUT
4 2 8	12 3 2

QUÀ TẶNG

Một công ty công nghệ có không quá 5000 nhân viên. Để giúp cho việc quản lý các nhân viên bằng hệ thống phần mềm, mỗi nhân viên được cấp một mã số ID là một số nguyên dương có tối đa 18 chữ số.

Trong buổi tiệc gala mừng kỉ niệm ngày thành lập, phần bốc thăm trúng quà như truyền thống được Ban giám đốc đổi mới bằng cách tặng quà cho các nhân viên có mã số ID là trung bình cộng ID của 2 nhân viên khác trong công ty.

Yêu cầu: Cho danh sách ID của các nhân viên trong công ty. Cho biết số lượng nhân viên được nhận quà tặng trong đêm gala.

Dữ liệu: Vào từ tập tin văn bản **GIFTS.INP** chứa dãy số nguyên là ID của tất cả nhân viên trong công ty, các ID cách nhau bởi khoảng trắng.

Kết quả: Ghi ra tập tin văn bản **GIFTS.OUT** một số nguyên là số nhân viên được nhận quà.

Ví dụ:

GIFTS . INP							GIFTS . OUT	
1	2	3	4	5	6	7	5	

TẶNG HOA

Biết mẹ rất thích hoa nên Cường đến cửa hàng hoa mua tặng mẹ một nhánh lan và một bông hồng (loại hoa mà mẹ Cường rất thích) bằng tiền tiết kiệm của mình nhân ngày của Mẹ. Cửa hàng có n nhánh lan đánh số từ 1 đến n với giá tiền tương ứng a_1, a_2, \dots, a_n và m bông hồng đánh số từ 1 đến m với giá tiền tương ứng b_1, b_2, \dots, b_m . Cường muốn mua một nhánh lan và một bông hồng với tổng giá tiền cao nhất (vì cậu nghĩ giá tiền càng cao thì hoa càng đẹp) nhưng không được vượt quá số tiền tiết kiệm S .

Yêu cầu: Hãy cho biết chỉ số của nhánh lan và chỉ số của bông hồng cần mua sao cho tổng giá tiền của hai bó hoa là lớn nhất và không vượt quá S . Nếu có nhiều cách chọn thì đưa ra một cách chọn một nhánh lan và một bông hồng có giá tiền chênh lệch nhau ít nhất.

Dữ liệu: Vào từ tập tin văn bản **FLOWERS.INP**

- Dòng đầu tiên chứa ba số nguyên n, m, S ($1 \leq n, m \leq 10^6; 1 \leq S \leq 10^9$)
- Dòng thứ hai chứa dãy số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)
- Dòng thứ ba chứa dãy số nguyên b_1, b_2, \dots, b_m ($1 \leq b_i \leq 10^6$)

Kết quả: Ghi ra tập tin văn bản **FLOWERS.OUT** trên một dòng hai chỉ số i và j tương ứng là chỉ số của nhánh lan và bông hồng cần mua. Trường hợp không tồn tại lời giải thì ghi hai số 0.

Ví dụ:

FLOWERS . INP	FLOWERS . OUT
4 6 10 5 3 7 2 4 1 8 9 5 6	1 5

MA TRẬN

Xét ma trận vuông kích thước $n \times n$, mỗi phần tử của ma trận là một số nguyên. Với mỗi hình vuông trù lên một ma trận vuông con người ta đánh giá độ đẹp của hình vuông. Gọi A là tổng các phần tử trên đường chéo chính của ma trận vuông con (đường chéo đi từ góc trên trái xuống góc dưới phải), B là tổng các phần tử trên đường chéo kia của ma trận vuông con. Độ đẹp của hình vuông là giá trị $A - B$.

Yêu cầu: Hãy xác định độ đẹp lớn nhất của các hình vuông có thể xét trong ma trận.

Dữ liệu: Vào từ tập tin văn bản **MATRIX.INP**

- Dòng đầu tiên chứa số nguyên $n (2 \leq n \leq 400)$,
- Dòng thứ i trong n dòng sau chứa n số nguyên xác định một dòng của ma trận. Các phần tử của ma trận có giá trị tuyệt đối không vượt quá 10^3 .

Kết quả: Đưa ra tập tin văn bản **MATRIX.OUT** một số nguyên – độ đẹp lớn nhất tìm được.

Ví dụ:

MATRIX . INP	MATRIX . OUT
3 -3 4 5 7 9 -2 1 0 -6	5

XÂU FIBONACCI 1

Công thức lặp có thể gặp với cả biểu thức xâu. Biểu thức xâu Fibonacci được xác định bằng công thức lặp $F_0 = a, F_1 = b, F_2 = F_0 + F_1, \dots, F_n = F_{n-2} + F_{n-1}, \dots$. Các xâu đầu tiên xác định theo công thức lặp này là **a, b, ab, bab, abbab, bababbab, abbabbababbab, ...**

Độ dài của xâu tăng lên rất nhanh. Vì vậy ta chỉ xét bài toán xác định một ký tự của một xâu trong dãy các xâu này.

Yêu cầu: Cho 2 số nguyên n và k . Hãy xác định ký tự thứ k của xâu F_n . Các ký tự trong F_n được đánh số bắt đầu từ 1.

Dữ liệu: Vào từ tập tin văn bản **FIB1.INP**

- Dòng đầu tiên chứa số nguyên T – số bộ dữ liệu test ($1 \leq T \leq 100$),
- Mỗi dòng trong T dòng sau chứa 2 số nguyên n và k ($0 \leq n \leq 45; 1 \leq k \leq \text{length}(F_n)$).

Kết quả: Ghi ra tập tin văn bản **FIB1.OUT**, kết quả mỗi test trên một dòng dưới dạng một ký tự.

Ví dụ:

FIB1 . INP	FIB1 . OUT
4	a
0 1	b
1 1	a
3 2	a
7 7	

XÂU FIBONACCI 2

Xâu Fibonacci thường được sử dụng để rèn luyện kỹ năng xử lý khi giới thiệu các giải thuật xử lý xâu.

Xét dãy xâu F_0, F_1, F_2, \dots xây dựng theo quy tắc sau:

- $F_0 = 'a'$
- $F_1 = 'b'$
- $F_n = F_{n-2} + F_{n-1}, n > 1$

Yêu cầu: Cho hai số nguyên n và k ($0 \leq n \leq 45$) (k không vượt quá độ dài xâu F_n). Hãy xác định số lượng ký tự a xuất hiện trong k ký tự đầu tiên của xâu F_n .

Dữ liệu: Vào từ tập tin văn bản **FIB2.INP**

- Dòng đầu tiên chứa số nguyên T – số lượng tests cần xử lý ($1 \leq T \leq 100$),
- Mỗi dòng trong T dòng sau chứa 2 số nguyên n và k .

Kết quả: Ghi ra tập tin văn bản **FIB2.OUT**, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên.

Ví dụ:

FIB2 . INP	FIB2 . OUT
4	1
0 1	0
1 1	1
3 2	3
7 7	

ĐOẠN CON

Cho dãy n số nguyên a_1, a_2, \dots, a_n và số nguyên dương k .

Yêu cầu: Tìm đoạn con liên tiếp gồm $\geq k$ số nguyên trong dãy, sao cho tổng các số nguyên thuộc đoạn là lớn nhất.

Dữ liệu: Vào từ tập tin văn bản **SUBSEQ.INP**

- Dòng 1 chứa hai số nguyên dương n, k ($1 \leq k \leq n \leq 10^6$)
- n dòng tiếp theo, dòng thứ i chứa một số nguyên a_i . ($|a_i| \leq 1000$)

Kết quả: Ghi tập tin văn bản **SUBSEQ.OUT** một số nguyên là tổng các giá trị đoạn con tìm được theo yêu cầu.

Ví dụ:

SUBSEQ . INP	SUBSEQ . OUT
8 3 -20 90 -30 -20 80 -70 -60 125	120

TỔNG ĐOẠN CON

Cho dãy số nguyên không âm: a_1, a_2, \dots, a_n . Xếp các số trong dãy theo thứ tự trên một vòng tròn theo chiều kim đồng hồ. Ta gọi một đoạn con của vòng tròn là một dãy các số liên tục theo chiều kim đồng hồ trên vòng tròn đó.

Yêu cầu: Hãy tìm một đoạn con có tổng bằng số nguyên S cho trước sao cho độ dài của đoạn con đó là ngắn nhất.

Dữ liệu: Vào từ tập tin văn bản **SUMSEQ.INP**

- Dòng đầu chứa hai số nguyên dương $n(n \leq 10^5)$ và S
- Dòng tiếp theo chứa dãy gồm n số nguyên không âm: $a_1, a_2, \dots, a_n(a_i \leq 10^9)$.

Kết quả: Ghi ra tập tin văn bản **SUMSEQ.OUT** hai số nguyên i và l , trong đó i là vị trí bắt đầu của đoạn con, l là độ dài của đoạn con. Nếu không tìm được dãy con nào, ghi ra một số 0 duy nhất.

Ví dụ:

SUMSEQ . INP	SUMSEQ . OUT
5 3 0 1 0 0 2	5 3

TỔNG THỨ K

Cho dãy số nguyên a_1, a_2, \dots, a_n . Xét tất cả cặp phần tử (a_i, a_j) với $1 \leq i < j \leq n$. Như vậy có tất cả $n * (n - 1)/2$ cặp phần tử.

Yêu cầu: Hãy tìm giá trị thứ k của $a_i + a_j$ nếu sắp xếp tất cả tổng này theo thứ tự không giảm.

Dữ liệu: Vào từ tập tin văn bản **KTHSUM.INP**

- Dòng đầu tiên chứa 2 số nguyên $n, k (2 \leq n \leq 10^5; 1 \leq k \leq n * (n - 1)/2)$.
- Dòng tiếp theo chứa dãy số nguyên $a_1, a_2, \dots, a_n (0 \leq a_i \leq 10^6)$.

Kết quả: Ghi ra tập tin văn bản **KTHSUM.OUT** một số nguyên là kết quả của bài toán.

Ví dụ:

KTHSUM . INP	KTHSUM . OUT
4 5 2 6 1 3	8

Giải thích: có tổng cộng 6 cặp phần tử $\{(a_1, a_2); (a_1, a_3); (a_1, a_4); (a_2, a_3); (a_2, a_4); (a_3, a_4)\}$ tương ứng với các giá trị $\{8; 3; 5; 7; 9; 4\}$. Giá trị thứ 5 của dãy sau khi sắp xếp không giảm là 8.

NGHIỆM NGUYÊN TỐ

Xét phương trình có dạng $x + y + z = K$ (K là một số nguyên dương). Bộ (x, y, z) được gọi là nghiệm nguyên tố của phương trình nếu $x + y + z = K$, $x \leq y \leq z$ và x, y, z đều là số nguyên tố.

Phương trình này có thể có vô số nghiệm, tuy nhiên, với mỗi K cho trước thì số nghiệm nguyên tố của phương trình hoàn toàn có thể xác định được.

Dữ liệu: Vào từ tập tin văn bản **EQUATION.INP** gồm nhiều dòng, mỗi dòng chứa một số nguyên dương K ($K < 5000$) là một trường hợp dữ liệu cần kiểm tra.

Kết quả: Ghi ra tập tin văn bản **EQUATION.OUT** gồm nhiều dòng, mỗi dòng ứng với dữ liệu của một trường hợp kiểm tra của dữ liệu vào là số bộ nghiệm nguyên tố của phương trình tương ứng.

Ví dụ:

EQUATION . INP	EQUATION . OUT
5	0
8	1

SỐ NGUYÊN TỔ ĐỐI XỨNG

Các số nguyên tố liệt kê theo thứ tự tăng dần 2, 3, 5, 7, 11, 13, ... tạo thành một dãy và đánh số bắt đầu từ 1. Gọi p_i là số nguyên tố thứ i , ta nói p_i là số nguyên tố đối xứng nếu nó bằng trung bình cộng của 2 số nguyên tố liền trước và liền sau nó. Nói cách khác p_i là số nguyên tố đối xứng nếu thỏa điều kiện:

$$p_i = \frac{p_{i-1} + p_{i+1}}{2}$$

Như vậy 10 số nguyên tố đối xứng đầu tiên là: 5, 53, 157, 173, 211, 257, 263, 373, 563, 593.

Yêu cầu: Cho số nguyên n . Cho biết n có phải là số nguyên tố đối xứng hay không.

Dữ liệu: Vào từ tập tin văn bản **SYMPRIME.INP**

- Dòng đầu tiên chứa số nguyên $t (1 \leq t \leq 10^5)$ – số lượng số n cần kiểm tra.
- Mỗi dòng trong t dòng tiếp theo chứa số nguyên $n (1 \leq n \leq 2 \times 10^7)$.

Kết quả: Ghi ra tập tin văn bản **SYMPRIME.OUT** gồm t dòng, mỗi dòng ghi thông báo YES hoặc NO là câu trả lời tương ứng với câu hỏi cho trong dữ liệu vào.

Ví dụ:

SYMPRIME . INP	SYMPRIME . OUT
3	NO
11	YES
5	YES
373	

PYTHAGORE

Định lí Pythagore do Pythagoras, nhà toán học người Hy Lạp sống vào khoảng thế kỉ thứ 6 trước công nguyên, phát minh. Ngày nay, định lí Pythagore được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như kiến trúc, xây dựng và đo lường. Những ứng dụng phổ biến của định lí Pythagore có thể kể ra là tính khoảng cách giữa 2 điểm trong mặt phẳng hoặc kiểm tra một tam giác có phải là tam giác vuông.

Định lí phát biểu như sau: Tam giác vuông có độ dài 3 cạnh a, b, c , trong đó c là độ dài cạnh huyền khi đó $a^2 + b^2 = c^2$.

Tí rất thích thú với tính chất này nhưng cậu ta chỉ quan tâm đến các tam giác vuông có độ dài ba cạnh a, b, c ($0 < a < b < c$) là số nguyên. Tí nhận thấy rằng, với cùng một chu vi thì có thể tồn tại nhiều tam giác vuông khác nhau có cạnh nguyên. Chẳng hạn với chu vi 60, tồn tại 2 tam giác vuông là (10,24,26) và (15,20,25). Còn với chu vi 12 chỉ tồn tại 1 tam giác vuông (3,4,5).

Yêu cầu: Cho trước chu vi P . Đếm số lượng tam giác vuông khác nhau có độ dài ba cạnh nguyên a, b, c ($a < b < c$) và chu vi là P .

Dữ liệu: Vào từ file văn bản **PYTHAGORE.INP**

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 5000$) – số bộ test.
- Dòng thứ i trong t dòng tiếp theo chứa số nguyên P_i ($1 \leq P_i \leq 5000$) – chu vi của các tam giác vuông thứ i .

Kết quả: Ghi ra file văn bản **PYTHAGORE.OUT** gồm t dòng, dòng thứ i ghi một số nguyên là số lượng tam giác vuông khác nhau có độ dài ba cạnh nguyên và chu vi P_i .

Ví dụ:

PYTHAGORE . INP	PYTHAGORE . OUT
2	1
12	2
60	

ĐƯỜNG ĐI TRÊN BẢNG

Cho bảng kích thước $n \times m$, mỗi ô của bảng chứa một số nguyên có giá trị từ 0 đến 9. Một đường đi trên bảng được gọi là hợp lệ nếu thỏa:

- Đi theo chiều ngang từ trái qua phải
- Đi theo chiều dọc từ trên xuống
- Đi theo đường chéo song song với đường chéo chính từ trên xuống.
- Độ dài đường đi không vượt quá 6 ô.
- Các chữ số trên đường đi tạo thành một số nguyên dương gọi là chi phí đường đi và chi phí đường đi là số nguyên tố.

Yêu cầu: Hãy tìm tất cả đường đi hợp lệ và tính tổng chi phí của tất cả đường đi hợp lệ.

Dữ liệu: Vào từ tập tin văn bản **TPATH.INP**

- Dòng đầu chứa 2 số nguyên $n, m (1 \leq n, m \leq 1500)$
- Mỗi dòng trong n dòng tiếp theo chứa dãy gồm m số nguyên (có giá trị từ 0 đến 9).

Kết quả: Ghi ra tập tin văn bản **TPATH.OUT** 2 số nguyên: số thứ nhất là số lượng đường đi hợp lệ - số thứ hai là tổng chi phí của tất cả đường đi hợp lệ.

Ví dụ:

TPATH . INP	TPATH . OUT
4 5 1 2 3 4 5 6 7 8 9 0 0 9 8 7 6 5 4 3 2 1	19 1547

Giải thích: có 19 đường đi hợp lệ: 2, 2, 3, 3, 5, 5, 7, 7, 17, 23, 43, 61, 67, 71, 79, 83, 89, 97, 883.

ĐỊNH ĐỀ BERTRAND

Định đề Bertrand phát biểu: với một số tự nhiên $n > 0$ luôn tồn tại một số nguyên tố p mà $n < p \leq 2n$.

Yêu cầu: Cho n , kiểm tra định đề Bertrand bằng cách đếm số lượng số nguyên tố nằm trong $[n + 1, 2n]$.

Dữ liệu: Vào từ tập tin văn bản **DDB.INP**

- Dòng đầu tiên chứa T là số bộ dữ liệu.
- T dòng sau, mỗi dòng tương ứng với một bộ dữ liệu là số nguyên $n \leq 10^6$.

Kết quả: Ghi ra tập tin văn bản **DDB.OUT** trên nhiều dòng, mỗi dòng ghi số lượng số nguyên tố trong đoạn tương ứng dữ liệu vào.

Ví dụ:

DDB . INP	DDB . OUT
2	1
1	1
3	

ĐIỀU KHIỂN ROBOT

Một robot xuất phát từ vị trí (0,0) mặt quay về hướng Bắc. Mỗi lần chỉ có một trong 4 lệnh chuyển động là G, L, R, B tương ứng tiến lên trên, tiến sang trái, tiến sang phải, quay lại phía sau một đơn vị theo hướng di chuyển hiện hành.

Yêu cầu: Cho dãy lệnh điều khiển Robot. Hãy xác định vị trí cuối cùng của Robot.

Dữ liệu: Vào từ tập tin văn bản **CROBOT.INP**

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 100$) – số lệnh điều khiển Robot.
- Dòng thứ hai chứa chuỗi độ dài n chỉ gồm các kí tự G, L, R, B – các lệnh điều khiển Robot.

Kết quả: Ghi ra tập tin văn bản **CROBOT.OUT** 2 số nguyên x, y – tọa độ cuối cùng của Robot.

Ví dụ:

CROBOT . INP	CROBOT . OUT
5 RLGGL	0 3

LÙA BÒ

Bờm được Phú Ông giao chăn đàn bò gồm n con được đánh số từ 1 đến n . Trong lúc Bờm đang mê ngủ, các con bò đã lên vào vườn trồng hoa quý của Phú Ông. Ngay khi Bờm phát hiện thì các con bò bắt đầu ăn hoa và Bờm quyết định phải đưa tất cả chúng về chuồng ngay lập tức.

Con bò thứ i có tốc độ ăn d_i cây hoa trong mỗi phút chờ đợi và nó đang ở cách chuồng một quãng đường mất t_i phút để di chuyển. Ở mỗi thời điểm, Bờm chỉ có thể đưa một con bò về chuồng. Bờm mất $2 \times t_i$ phút để đưa con bò thứ i về chuồng và quay trở lại.

Bờm đang lo lắng vì không biết đưa cả đàn bò về chuồng thế nào để tổng số hoa quý của Phú Ông bị thiệt hại là ít nhất.

Yêu cầu: Cho 2 dãy số nguyên t_i và d_i tương ứng với thời gian di chuyển về chuồng và tốc độ ăn hoa của con bò thứ i . Hãy tính số cây hoa bị thiệt hại ít nhất sau khi cả đàn bò về chuồng.

Dữ liệu: Vào từ tập tin văn bản **COWS.INP**

- Dòng đầu tiên chứa số nguyên n ($2 \leq n \leq 10^5$)
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên t_i, d_i ($1 \leq t_i \leq 2 \times 10^6$; $1 \leq d_i \leq 100$).

Kết quả: Ghi ra tập tin văn bản **COWS.OUT** một số nguyên là số cây hoa ít nhất bị đàn bò ăn.

Ví dụ:

COWS . INP	COWS . OUT
6 3 1 2 5 2 3 3 2 4 1 1 6	86

XÓA CHỮ SỐ

Cho S là xâu gồm n ký tự ($n < 10^6$) là các chữ số là biểu diễn thập phân của một số nguyên dương. Hãy xóa trong xâu S đi đúng k chữ số ($k < n$) để thu được xâu T là biểu diễn thập phân của một số nguyên dương với giá trị lớn nhất có thể.

Dữ liệu: Vào từ tập tin văn bản **DIGIT.INP**

- Dòng 1 chứa xâu S .
- Dòng 2 chứa số nguyên dương k .

Kết quả: Ghi ra tập tin văn bản **DIGIT.OUT** xâu T tìm được.

Ví dụ:

DIGIT . INP	DIGIT . OUT
7918256 3	9856