

BÀI TẬP LẬP TRÌNH**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

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	

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:

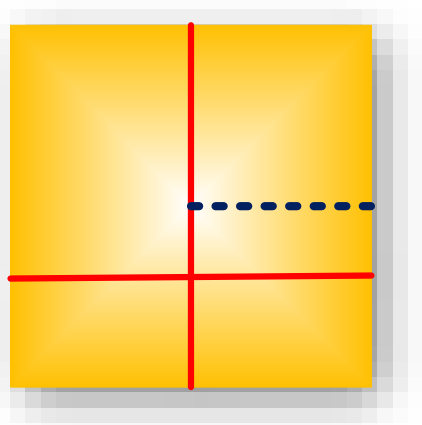
KTHSUM . INP	KTHSUM . OUT
4 5 2 6 1 3	8

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.

MIẾNG BÁNH VUÔNG

Steve được tặng một chiếc bánh ga tô sinh nhật hình chữ nhật kích thước $m \times n$. Steve rất thích hình vuông và dự định cắt nó ra thành các miếng hình vuông. Để làm được việc đó Steve tưởng tượng chiếc bánh như một hình chữ nhật có tọa độ góc dưới trái là $(0, 0)$ và tọa độ góc trên phải là (m, n) .

Đang mãi suy nghĩ nên Steve không nhận thấy em mình đã nhanh tay cắt thực hiện k lần cắt, mỗi lát cắt song song với cạnh của chiếc bánh và đi suốt toàn bộ bánh. Điều này làm Steve hết sức buồn rầu chán nản và quyết định chỉ cắt một miếng hình vuông lớn nhất đặt lên bàn.



Yêu cầu: Cho n, m, k và các lát cắt t_i, v_i , trong đó $t_i = 0$ xác định lát cắt $x = v_i (0 \leq x \leq n)$, $t_i = 1$ xác định lát cắt $y = v_i (0 \leq y \leq m)$. Hãy xác định độ dài d cạnh hình vuông lớn nhất có thể cắt.

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

- Dòng đầu tiên chứa 3 số nguyên $n, m, k (1 \leq n, m \leq 10^9, 0 \leq k \leq 10^5)$
- Dòng thứ i trong k dòng sau chứa 2 số nguyên t_i, v_i .

Kết quả: Ghi ra tập tin văn bản **MAXPIECE.OUT** số nguyên d .

Ví dụ:

MAXPIECE . INP	MAXPIECE . OUT
10 10 2 1 5 0 3	5

CHƠI ĐÀN GHI TA

Trong cuộc thi sáng tạo robot, Steve đã trình diễn robot chơi ghi-ta của mình, một robot, theo lời mô tả của tác giả “có đến hàng tỷ ngón tay”.

Đàn ghi ta có 6 dây đánh số từ 1 đến 6 và có p phím đánh số từ 1 đến p . Khi chơi nhạc người ta gảy dây và bấm phím để có các giai điệu khác nhau. Với mỗi dây, nếu có nhiều phím cùng được bấm thì âm điệu sẽ được quyết định bởi phím có số cao nhất. Ví dụ, âm điệu khi gảy dây số 3 và bấm đồng thời các phím 5 và 7 sẽ giống như khi ta gảy dây này và chỉ bấm phím số 7.

Điều khó khăn nhất khi biểu diễn một bài nhạc là điều khiển cường độ tức phải gảy dây nào và bấm phím nào, còn điều khiển trường độ (thời gian phát một nốt) không phải là quá khó về phương diện kỹ thuật.

Chương trình điều khiển cường độ của Steve được tối ưu hóa theo hướng giảm số lần chuyển tay bấm phím xuống còn ít nhất.

Xét bản nhạc có n nốt, nốt thứ i được cho bởi cặp giá trị nguyên (s_i, f_i) , trong đó s_i – dây cần gảy và f_i – phím cần bấm ($1 \leq s_i \leq 6; 1 \leq f_i \leq p$). Bản nhạc cần chơi theo trình tự thực hiện lần lượt từ nốt 1 đến nốt n . Ví dụ, với $n = 5, p = 15$ và bản nhạc là $(2,8), (2,10), (2,12), (2,10), (2,5)$, thao tác bấm phím của robot là bấm phím 8, bấm phím 10, bấm phím 12, nhả phím 12, nhả phím 10, nhả phím 8 và bấm phím 5, tất cả là 7 thao tác, thực hiện trên dây số 2.

Yêu cầu: Cho n, p và các s_i, f_i ($1 \leq n \leq 5 \times 10^5; 2 \leq p \leq 3 \times 10^5$). Hãy xác định số thao tác bấm phím ít nhất cần thực hiện.

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

- Dòng đầu tiên chứa hai số nguyên n và p ,
- Dòng thứ i trong n dòng sau chứa hai số nguyên s_i và f_i .

Kết quả: Ghi ra tập tin văn bản **GUITAR.OUT** một số nguyên – số thao tác bấm phím ít nhất cần thực hiện.

Ví dụ:

GUITAR . INP	GUITAR . OUT
5 15 2 8 2 10 2 12 2 10 2 5	7

CỬA SỔ ỨNG DỤNG

Khi làm việc trong môi trường Windows, ta thường mở nhiều cửa sổ ứng dụng. Mỗi cửa sổ là một hình chữ nhật có các cạnh song song với các cạnh của màn hình gồm một số ô trong lưới $10^4 \times 10^4$ ô vuông đơn vị. Các dòng của lưới được đánh số từ 0 từ trên xuống, các cột được đánh số từ 0 từ trái sang phải. Mỗi cửa sổ được xác định bởi tọa độ ô trái trên và ô phải dưới. Nếu bấm chuột vào ô phải trên thì cửa sổ sẽ đóng lại.

Trong quá trình mở cửa sổ ứng dụng, cửa sổ mở sau có thể che một phần cửa sổ mở trước và một cửa sổ chỉ có thể đóng được nếu ô phải trên của nó không bị che.

Yêu cầu: Cho dãy n cửa sổ đánh số từ 1 được mở theo thứ tự đó, cần phải dùng ít nhất bao nhiêu lần đóng cửa sổ để có thể đóng được cửa sổ thứ 1.

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

- Dòng đầu tiên ghi số nguyên $n \leq 1000$.
- Dòng thứ i trong n dòng tiếp theo ghi 4 số $x_{i1}, y_{i1}, x_{i2}, y_{i2}$ với ý nghĩa (x_{i1}, y_{i1}) là tọa độ ô trái trên và (x_{i2}, y_{i2}) là tọa độ ô phải dưới của cửa sổ thứ i .

Kết quả: Ghi ra tập tin văn bản **WINS.OUT**

- Dòng đầu ghi số nguyên S là số lần đóng cửa sổ.
- Dòng thứ hai ghi S số hiệu của cửa sổ lần lượt đóng.

Ví dụ:

WINS . INP	WINS . OUT
3	2
2 3 7 7	2 1
4 1 9 5	
3 2 6 6	

DÃY ĐẢO TRỊ

Xét dãy số vô hạn A , các phần tử có giá trị 0,1 được đánh chỉ số từ 1 trở đi. Dãy A được xây dựng bằng cách lặp lại vô hạn 2 thao tác sau từ dãy ban đầu gồm 1 phần tử $\{0\}$:

- Tạo dãy B có độ dài bằng dãy A , trong đó $B[i] = 1 - A[i]$. Chẳng hạn với $A = \{0,1\}$ thì $B = \{1,0\}$.
- Dãy A được cập nhật bằng cách ghép dãy B vào sau. Chẳng hạn $A = \{0,1\}$ thì sau phép ghép ta có $A = \{0,1,1,0\}$.

Bảng mô tả quá trình xây dựng dãy A như sau:

Lần	Dãy A	Dãy B	Dãy A sau thao tác
1	0	1	0,1
2	0,1	1,0	0,1,1,0
3	0,1,1,0	1,0,0,1	0,1,1,0,1,0,0,1
4	0,1,1,0,1,0,0,1	1,0,0,1,0,1,1,0	0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0
...			

Yêu cầu: Cho biết giá trị phần tử thứ n của dãy.

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

- Dòng đầu chứa số nguyên $t (t \leq 10^5)$ – số bộ test.
- Mỗi dòng trong t dòng tiếp theo chứa số nguyên $n (1 \leq n \leq 10^{18})$ – vị trí của phần tử cần tìm.

Kết quả: Ghi ra tập tin văn bản **INVERSEQ.OUT** gồm t dòng là giá trị của phần tử ở vị trí tương ứng.

Ví dụ:

INVERSEQ . INP	INVERSEQ . OUT
3	1
2	1
5	0
10	

XÂY THÁP

Cho n khối lập phương, khối thứ i có kích thước a_i . Người ta sử dụng các khối lập phương để xây tháp bằng cách xếp chồng lên nhau theo nguyên tắc: khối lập phương nằm trên phải có kích thước nhỏ hơn khối nằm bên dưới.

Yêu cầu: Sử dụng các khối lập phương theo thứ tự đã cho để xây dựng tháp. Mỗi khối lập phương có thể đặt lên đầu của một trong các tháp hiện có hoặc tạo thành tháp mới. Hãy tìm số tháp ít nhất xây dựng được.

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

- Dòng đầu tiên chứa số nguyên $n (1 \leq n \leq 2 \times 10^5)$ – số lượng khối lập phương.
- Dòng tiếp theo chứa dãy gồm n số nguyên $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$ – kích thước các khối lập phương.

Kết quả: Ghi ra tập tin văn bản **TOWERS.OUT** một số nguyên là số tháp ít nhất xây dựng được.

Ví dụ:

TOWERS . INP	TOWERS . OUT
5 3 8 2 1 5	2

SÁNG TÁC THƠ

Bờm sáng tác một bài thơ có quy tắc gieo vần như sau: nếu 2 câu cách nhau đúng m dòng thì k ký tự cuối của các câu đó phải giống nhau, bao gồm cả dấu cách, giống nhau cả cách viết hoa hay thường.

Bờm đã sáng tác được n dòng thơ. Bờm đề nghị Phú Ông sáng tác tiếp. Bờm có giải thích cho Phú Ông về quy tắc gieo vần nhưng do tuổi cao nên khi Bờm rời đi thì Phú Ông đã quên mất giá trị k .

Không muốn bị mất mặt với Bờm, việc đầu tiên của Phú Ông là dựa vào n dòng thơ đã được sáng tác trước đó để xác định giá trị k lớn nhất thỏa mãn quy tắc gieo vần.

Yêu cầu: Hãy xác định giá trị k lớn nhất mà Phú Ông sử dụng khi kéo dài bài thơ.

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

- Dòng đầu tiên chứa 2 số nguyên n, m ($1 \leq m \leq n \leq 1000$).
- Mỗi dòng trong n dòng sau chứa một dòng thơ dưới dạng xâu độ dài không quá 1000 ký tự chữ cái la tinh (hoa hoặc thường) và dấu cách. Xâu không bắt đầu và không kết thúc bằng dấu cách.

Kết quả: Ghi ra tập tin văn bản **LYRICS.OUT** một số nguyên – giá trị k tìm được.

Ví dụ:

LYRICS . INP	LYRICS . OUT
2 1 Vietnam Vietnam go go go Vietnam Vietnam lets go	3

TUYỂN BAY

Steve sinh hoạt ở câu lạc bộ mô hình máy bay. Hôm nay là ngày kiểm tra kỹ thuật lái máy bay mô hình. Chương trình bay tự động đã được cài sẵn trong máy bay, tại thời điểm i máy bay sẽ bay ở độ cao h_i .

Steve phải dùng thiết bị cầm tay điều khiển máy bay bay ở một độ cao ổn định luôn luôn là p . Do điều kiện địa hình và khoảng cách từ nơi điều khiển đến máy bay nên ở thời điểm i cứ thay đổi mỗi đơn vị độ cao (lên hoặc xuống) so với giá trị định sẵn phải mất a_i đơn vị năng lượng.

Yêu cầu: Tìm cách đảm bảo độ cao ổn định của máy bay với chi phí năng lượng nhỏ nhất.

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

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^6$).
- Dòng thứ ba chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$).

Kết quả: Đưa ra tập tin văn bản **FLY.OUT** trên một dòng số nguyên p và chi phí năng lượng cần thiết để điều khiển. Nếu tồn tại nhiều độ cao khác nhau cùng thỏa mãn thì đưa ra độ cao nhỏ nhất.

Ví dụ:

FLY . INP	FLY . OUT
6 6 7 8 8 7 7 10 6 3 1 1 4	7 14

LŨY THỪA

Với số x cho trước có nhiều cách để tính x^n , trong đó n là một số nguyên ($1 \leq n \leq 10^{18}$). Ví dụ, với $n = 5$ ta thể tính x^5 theo các cách:

Cách 1	Cách 2
$r = x * x$	$r = x * x$
$r = r * x$	$r = r * r$
$r = r * x$	$r = r * x$
$r = r * x$	

Theo cách 1 ta phải thực hiện 4 phép tính nhân, cách 2 chỉ đòi hỏi phải thực hiện 3 phép tính nhân.

Yêu cầu: Cho số nguyên n . Hãy xác định số phép tính nhân ít nhất cần thực hiện để tính x^n .

Dữ liệu: Vào từ tập tin văn bản **POWER.INP** gồm một số nguyên n .

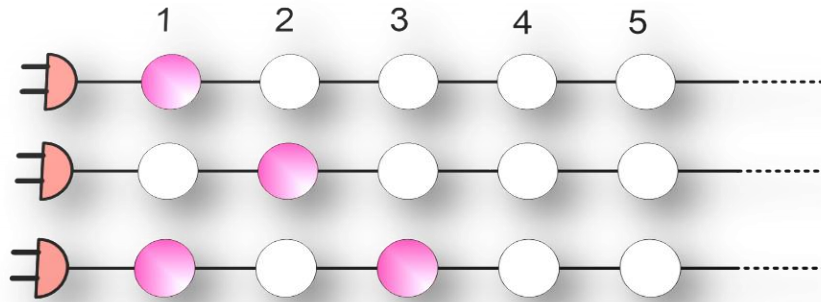
Kết quả: Ghi ra tập tin văn bản **POWER.OUT** một số nguyên – số phép tính nhân ít nhất cần thực hiện.

Ví dụ:

POWER . INP	POWER . OUT
5	3

ĐÈN TRANG TRÍ

Đường phố chính của thành phố trong ngày lễ được trang trí bằng một dây đèn nhấp nháy duy nhất dài vô hạn, quấn vào cây và các biểu tượng chào mừng chạy dọc suốt tuyến phố. Đèn nhấp nháy theo quy tắc sau: khi mới cắm điện duy nhất chỉ có đèn sát ô điện sáng. Cứ mỗi giây mỗi bóng đèn kiểm tra, nếu cạnh nó có đúng một bóng đèn sáng thì bóng đèn này cũng sẽ sáng ở giây tiếp theo, trong trường hợp ngược lại – sẽ tắt. Bóng đèn đầu tiên chỉ có một bóng ở cạnh nó.



Yêu cầu: Cho số nguyên $n (1 \leq n \leq 10^{18})$. Hãy xác định xem ở giây thứ n có bao nhiêu bóng đèn sáng.

Dữ liệu: Vào từ tập tin văn bản **LIGHT.INP** chứa số nguyên n .

Kết quả: Ghi ra tập tin văn bản **LIGHT.OUT** một số nguyên – số bóng đèn sáng.

Ví dụ:

LIGHT . INP	LIGHT . OUT
5	2

ĐUA XE TÍNH ĐIỂM

Đua xe là một môn thể thao hấp dẫn nhiều người hâm mộ. Tuy nhiên, để tăng thêm phần kịch tính cho cuộc đua, người ta bố trí nhiều đường đua khác nhau. Một trong những cách tổ chức hấp dẫn nhiều cua-rơ nhất là đua xe tính điểm. Theo cách tổ chức này, trường đua là một vùng sân hình chữ nhật được chia thành lưới gồm n dòng, mỗi dòng chứa m ô; các dòng được đánh số từ 1 đến n kể từ Bắc xuống Nam, các cột được đánh số từ 1 đến m kể từ Tây sang Đông. Trong mỗi ô, Ban Tổ Chức có ghi một con số nguyên không âm. Khi xe đi qua ô nào thì số của ô đó sẽ được cộng vào cho cua-rơ – lúc xuất phát điểm của cua-rơ là 0. Xe đua là xe chỉ có thể chạy theo 3 hướng: Tây \rightarrow Đông (ký hiệu F), Tây Nam \rightarrow Đông Bắc (U) và Tây Bắc \rightarrow Đông Nam (D). Cua-rơ có thể xuất phát từ bất cứ ô nào ở trên cạnh phía Tây của trường đua và chạy theo các hướng trên. Việc đổi hướng của xe chỉ được thực hiện tại tâm của mỗi ô. Cua-rơ sẽ kết thúc lượt chạy khi đến được một trong số các ô trên biên phía Đông của trường đua. Khi đó, tiền thưởng cho cua-rơ sẽ bằng chính số điểm mà anh ta đạt được. Các xe chạy vượt qua khỏi các biên phía Bắc và phía Nam trường đua sẽ bị coi là phạm quy và nhận được số tiền thưởng là 0.

Yêu cầu: Cho trước số các thông số của trường đua, hãy xác định chỉ số dòng x của ô xuất phát ở cột 1 và các hướng đi cho xe tại mỗi ô đi qua sao cho số tiền thưởng T mà cua-rơ nhận được là lớn nhất.

Dữ liệu: Vào được cho từ tập tin văn bản **RACE.INP**

- Dòng đầu chứa số 2 số nguyên n, m ($n, m \leq 1000$).
- Trong n dòng tiếp theo, mỗi dòng chứa m số là các số trong các ô trên trường đua ở vị trí tương ứng; giá trị của các số này không vượt quá 10^6 .

Kết quả: Ghi ra tập tin văn bản **RACE.OUT**

- Dòng đầu ghi 2 số nguyên dương x, T .
- Dòng thứ 2 chứa chuỗi gồm $m - 1$ ký tự chỉ hướng đi tiếp theo của xe tại các ô trên đường chạy theo đúng trình tự.

Ví dụ:

RACE . INP					RACE . OUT	
4	4				3	42
1	10	11	8		DFU	
9	2	7	12			
8	6	3	11			
5	13	10	4			

HÌNH VUÔNG CỰC ĐẠI

Cho một bảng kích thước $m \times n$, được chia thành lưới ô vuông đơn vị m dòng n cột. Các dòng của bảng được đánh số từ 1 tới m theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số từ 1 tới n theo thứ tự từ trái qua phải. Ô (i, j) của bảng chứa một ký tự c_{ij} trong đó c_{ij} có thể là dấu “*” (dấu hoa thị) hoặc dấu “.” (dấu chấm). Có ít nhất một ô chứa dấu “*.”.

Yêu cầu: Hãy tìm một hình vuông chiếm trọn một số ô của bảng thoả mãn các điều kiện sau:

- Hình vuông chỉ gồm các ô chứa dấu “*.”
- Cạnh hình vuông song song với cạnh bảng
- Kích thước hình vuông là lớn nhất có thể.

Dữ liệu: vào từ tập tin văn bản **SQUARE.INP**

- Dòng đầu ghi hai số m, n cách nhau một dấu cách ($m, n \leq 1000$)
- m dòng tiếp theo, dòng thứ i chứa n ký tự liên nhau, ký tự thứ j là c_{ij}

Kết quả: ghi ra tập tin văn bản **SQUARE.OUT** một số nguyên duy nhất là độ dài cạnh hình vuông tìm được.

Ví dụ:

SQUARE . INP	SQUARE . OUT
5 6 ..*.*. *.**** *****. *****. ...*..	3

SƠN NHÀ

Có n căn nhà cần sơn. Căn nhà i được sơn bằng một trong 3 màu Xanh, Hồng, Vàng với mức giá tương ứng là a_{i1}, a_{i2}, a_{i3} .

Yêu cầu: Tìm cách sơn màu cho n ngôi nhà sao cho hai căn nhà cạnh nhau không được sơn cùng màu và tổng chi phí sơn là ít nhất.

Dữ liệu: Vào từ tập tin văn bản **PAINT.INP** chứa số nguyên T là số bộ test ($T \leq 100$). Mỗi bộ test có dạng

- Dòng đầu chứa số nguyên dương $n(1 \leq n \leq 20)$ là số ngôi nhà,
- Dòng thứ j trong n dòng sau chứa 3 số nguyên không âm a_{j1}, a_{j2}, a_{j3} là chi phí sơn ngôi nhà j bằng các màu Xanh, Hồng, Vàng tương ứng ($0 \leq a_{jk} \leq 1000$).

Kết quả: Ghi ra tập tin văn bản **PAINT.OUT** gồm T dòng, dòng thứ i là kết quả của bộ test thứ i tương ứng trong dữ liệu vào.

Ví dụ:

PAINT . INP	PAINT . OUT
2	137
4	96
13 23 12	
77 36 64	
44 89 76	
31 78 45	
3	
26 40 83	
49 60 57	
13 89 99	

Giải thích test 2: ngôi nhà 1 và 3 sơn màu Xanh, ngôi nhà 2 sơn màu Vàng.

Giới hạn

- 30% số test đầu tiên có $n \leq 3$
- 30% số test tiếp theo có $3 < n \leq 15, T = 1$
- 40% số test cuối có $n \leq 20, T \leq 100$.

TRÍ THÔNG MINH CỦA LOÀI KHỦNG LONG

Một số người cho rằng những con khủng long càng to thì càng thông minh. Để bác bỏ điều này bạn phải khảo sát một nhóm khủng long và chỉ ra rằng trong đó tồn tại một nhóm gồm nhiều con khủng long nhất sao cho trọng lượng của chúng tăng dần nhưng chỉ số IQ của chúng thì giảm dần.

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

- Dòng đầu tiên chứa số nguyên dương $n \leq 1000$
- Trong n dòng tiếp theo, dòng thứ i chứa 2 số nguyên dương w_i và s_i ($w_i, s_i \leq 10^5$) là trọng lượng và chỉ số IQ của con khủng long thứ i tương ứng ($1 \leq i \leq n$).

Kết quả: Ghi ra tập tin văn bản **DINOSAUR.OUT**:

- Dòng đầu tiên chứa số nguyên m là số lượng khủng long nhiều nhất tìm được.
- Dòng tiếp theo ghi m số nguyên dương $idx[1], idx[2], \dots, idx[m]$ là chỉ số của các con khủng long sao cho theo thứ tự này thỏa điều kiện

$$w[idx[1]] < w[idx[2]] < \dots < w[idx[m]] \text{ và } s[idx[1]] > s[idx[2]] > \dots > s[idx[m]]$$

Nếu bài toán tồn tại nhiều nghiệm thì chỉ tìm ra một nghiệm trong số đó.

Ví dụ:

DINOSAUR . INP	DINOSAUR . OUT
9 6008 1300 6000 2100 500 2000 1000 4000 1100 3000 6000 2000 8000 1400 6000 1200 2000 1900	4 4 5 9 7

ĐOẠN GỐI

Cho n đoạn thẳng với các điểm đầu a_i và điểm cuối b_i là những số nguyên trong khoảng -1000 đến 1000, $a_i < b_i$. Hãy tìm tối đa k đoạn thẳng gối nhau liên tiếp. Hai đoạn thẳng $[a, b]$ và $[c, d]$ được gọi là gối nhau nếu xếp chúng trên cùng một trục số thì điểm đầu đoạn này trùng với điểm cuối của đoạn kia, nghĩa là $b = c$ hoặc $d = a$.

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

- Dòng đầu tiên chứa số nguyên dương n ($n \leq 1000$)
- Dòng thứ i trong n dòng tiếp theo, mỗi dòng chứa số nguyên a_i, b_i biểu diễn điểm đầu và điểm cuối của đoạn thứ i

Kết quả: Ghi ra tập tin văn bản **OVERLAP.OUT**

- Dòng đầu ghi số nguyên k
- k dòng tiếp theo, mỗi dòng ghi chỉ số của các đoạn thẳng gối nhau liên tiếp

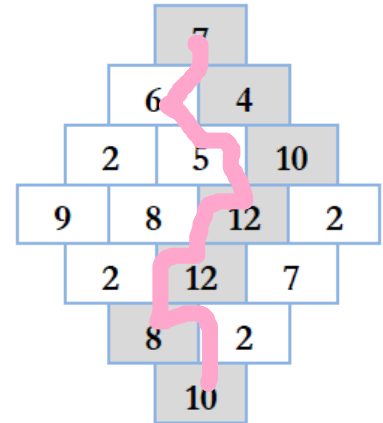
Ví dụ:

OVERLAP . INP	OVERLAP . OUT
5	3
2 7	2
1 3	4
7 9	5
3 4	
4 5	

KHI TRỘM CHUỐI

Một con khi đã tìm được cách để đột nhập vào thánh địa chuỗi gồm các lưới ô vuông bố trí theo dạng hình kim cương như minh họa. Chú khi chỉ có thể nhảy xuống các ô vuông kề cạnh bên dưới ô vuông nó đang đứng và chú khi sẽ lấy sạch lượng chuỗi ở những ô vuông mà chú đến được.

Chú khi sẽ tiến vào thánh địa chuỗi từ cửa vào là ô nằm trên cùng và sẽ thoát ra ngoài theo cửa ra là ô nằm dưới cuối của thánh địa. Chú khi muốn tìm một đường đi trong thánh địa sao cho có thể lấy được lượng chuỗi nhiều nhất có thể.



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

- Dòng đầu tiên chứa số nguyên $T (T \leq 50)$ tương ứng số bộ test.
- Mỗi bộ test nằm trên nhiều dòng. Dòng đầu tiên chứa số nguyên $n (1 \leq n \leq 100)$. Tiếp theo gồm $2 \times n - 1$ dòng mô tả số lượng chuỗi trong mỗi ô vuông tương ứng
 - + Dòng thứ $i (1 \leq i \leq n)$ trong n dòng tiếp theo chứa i số nguyên không vượt quá 2^{15}
 - + Dòng thứ $j (1 \leq j \leq n - 1)$ trong $n - 1$ dòng tiếp theo chứa $n - j$ số nguyên không vượt quá 2^{15}

Kết quả: Ghi ra tập tin văn bản **BANANAS.OUT** gồm T dòng, mỗi dòng tương ứng với số chuỗi thu được nhiều nhất ở từng bộ test.

Ví dụ:

BANANAS . INP	BANANAS . OUT
2	63
4	5
7	
6 4	
2 5 10	
9 8 12 2	
2 12 7	
8 2	
10	
2	
1	
2 3	
1	

DÂY HÌNH CHỮ NHẬT LỒNG NHAU

Trên mặt phẳng tọa độ cho n hình chữ nhật (được đánh số từ 1 tới n) có các cạnh song song với hệ trục tọa độ. Hình chữ nhật thứ i được cho bởi 4 số nguyên dương $x_{i1}, y_{i1}, x_{i2}, y_{i2}$ trong đó (x_{i1}, y_{i1}) là tọa độ đỉnh trái dưới, còn (x_{i2}, y_{i2}) là tọa độ đỉnh phải trên. Ta nói rằng hình chữ nhật thứ i nằm trong hình chữ nhật thứ j nếu trên mặt phẳng tọa độ, mọi điểm của hình chữ nhật i đều thuộc hình chữ nhật j .

Yêu cầu: Với n hình chữ nhật cho trước, hãy tìm k hình chữ nhật với chỉ số i_1, i_2, \dots, i_k sao cho hình i_1 nằm trong hình i_2 , hình i_2 nằm trong hình i_3 , ..., hình i_{k-1} nằm trong hình i_k và k là lớn nhất.

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

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$)
- n dòng tiếp theo, dòng thứ i chứa bốn số nguyên dương $x_{i1}, y_{i1}, x_{i2}, y_{i2}$ có giá trị không vượt quá 200.

Kết quả: Ghi ra tập tin văn bản **NESTED.OUT** số k tìm được.

Ví dụ:

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

TỔNG CHẴN LỚN NHẤT

Cho lưới hình chữ nhật gồm m hàng, các hàng đánh số từ 1 đến m và n cột, các cột đánh số từ 1 đến n , giao của hàng i cột j chứa số nguyên dương a_{ij} . Từ ô (i, j) chỉ có thể di chuyển đến các ô $(i, j + 1)$ và $(i + 1, j)$. Giá trị của lưới là tổng giá trị các ô là số chẵn thuộc một đường đi từ ô $(1, 1)$ đến ô (m, n) .

Yêu cầu: Xác định giá trị lớn nhất của lưới.

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

- Dòng đầu chứa 2 số m và n ($m, n \leq 1000$),
- Mỗi dòng trong m dòng tiếp theo chứa n số nguyên dương không vượt quá 10^9 mô tả dữ liệu của lưới hình chữ nhật.

Kết quả: Ghi ra tập tin văn bản **MAXSUM.OUT** số nguyên là theo yêu cầu của bài toán.

Ví dụ:

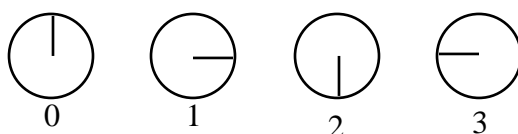
MAXSUM . INP	MAXSUM . OUT
2 3 2 3 4 3 2 2	8

CLOCKS

Cho 9 chiếc đồng hồ được sắp thành ma trận 3×3 như sau:

A	B	C
D	E	F
G	H	I

Mỗi đồng hồ ở 1 trong 4 trạng thái sau:



Cho 9 phép biến đổi, mỗi biến đổi tác động lên một số đồng hồ nhất định và quay các kim một góc 90° .

Phép biến đổi	Các đồng hồ bị tác động
1	A, B, D, E
2	A, B, C
3	B, C, E, F
4	A, D, G
5	B, D, E, F, H
6	C, F, I
7	D, E, G, H
8	G, H, I
9	E, F, H, I

Yêu cầu: Cho trước trạng thái ban đầu của các đồng hồ. Hãy thực hiện một số ít nhất các phép biến đổi trên để đưa tất cả đồng hồ về trạng thái 0

Dữ liệu: Vào từ tập tin văn bản **CLOCKS.INP** gồm 3 dòng, mỗi dòng gồm 3 số nguyên có giá trị từ 0 đến 3 mô tả trạng thái ban đầu của các đồng hồ được sắp thành ma trận 3×3

Kết quả: Ghi ra tập tin văn bản **CLOCKS.OUT**

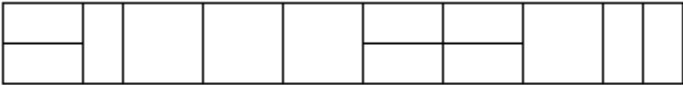
- Dòng đầu tiên ghi số phép biến đổi tối thiểu
- Các dòng tiếp theo ghi chỉ số các phép biến đổi cần thực hiện theo thứ tự bất kỳ, mỗi số trên một dòng
- Trường hợp không tìm được lời giải thì ghi NO SOLUTION

Ví dụ:

CLOCKS . INP	CLOCKS . OUT
0 2 3	3
2 1 2	5
3 2 0	7

LÁT VIỀN

Đường viền trang trí ở nền nhà có kích thước $2 \times n$ được lát bằng 2 loại gạch: loại kích thước 1×2 và loại 2×2 . Hãy xác định số cách lát khác nhau có thể thực hiện.



Dữ liệu: Vào từ tập tin văn bản **TILING.INP** gồm nhiều dòng, mỗi dòng chứa một số nguyên dương n ($n \leq 50$).

Kết quả: Ghi ra tập tin văn bản **TILING.OUT** các kết quả tìm được, mỗi số trên một dòng.

Ví dụ:

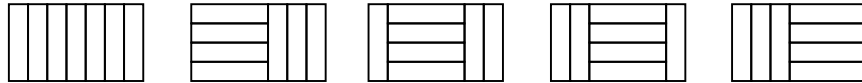
TILING . INP	TILING . OUT
2	3
8	171
12	2731

LÁT SÂN NHÀ

Bờm và Phú Ông đang ngồi trước sân tận hưởng những ngọn gió ít ỏi của buổi chiều hè. Cả hai cảm thấy buồn chán và Phú Ông nảy ra một trò chơi và treo thưởng nếu Bờm giải được sẽ thưởng cho một gói xôi, món ăn yêu thích của Bờm.

Sân nhà Phú Ông là một hình chữ nhật kích thước $4 \times n$ ô vuông đơn vị. Nhà Phú Ông giàu có cũng đang chất một đồng gạch loại 1×4 với số lượng không biết bao nhiêu mà kể. Phú Ông yêu cầu Bờm hãy cho biết số cách khác nhau để lát kín mặt sân bằng những viên gạch đó. Tuy nhiên, Phú Ông sợ Bờm có thể dễ dàng giải câu đố của mình nên đã yêu cầu Bờm đếm số lượng số nguyên tố có giá trị không vượt quá số cách lát mặt sân.

Chẳng hạn với $n = 7$, ta có 5 cách lát khác nhau như sau



Khi đó có 3 số nguyên tố không vượt quá 5 là 2,3,5.

Yêu cầu: Cho số nguyên $n(1 \leq n \leq 40)$. Hãy giúp Bờm đếm số lượng số nguyên tố thỏa yêu cầu của Phú Ông.

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

- Dòng đầu tiên chứa số nguyên dương $T(1 \leq T \leq 20)$ – số câu đố của Phú Ông.
- Mỗi dòng trong T dòng tiếp theo chứa số nguyên $n(1 \leq n \leq 40)$.

Kết quả: Ghi ra tập tin văn bản **YARD.OUT** gồm T dòng, mỗi dòng là kết quả của câu đố tương ứng.

Ví dụ:

YARD . INP	YARD . OUT
1 7	3

THÁP XU

Alice và Bob chơi trò bốc xu từ tháp được xây dựng bởi n đồng xu. Hai bạn chọn hai số nguyên dương khác nhau K và L . Hai người lần lượt đi. Alice đi trước. Mỗi người, khi đến lượt mình, được bốc khỏi tháp 1, K hoặc L xu. Ai bốc được đồng xu (hoặc các đồng xu) cuối cùng là thắng. Sau rất nhiều lần chơi, Alice nhận thấy rằng có những trường hợp mình chắc chắn thắng không phụ thuộc vào cách đi của Bob, ngược lại, có trường hợp dù đi thế nào Bob vẫn thắng. Trước ván chơi mới Alice nóng lòng muốn biết mình có thắng được hay không.

Yêu cầu: Cho n , K và L . Hãy xác định Alice hay Bob sẽ thắng.

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

- Dòng đầu tiên chứa 3 số nguyên K , L và m , trong đó m – số ván chơi ($1 < K < L < 10$; $3 < m < 50$).
- Dòng thứ hai chứa m số nguyên n_1, n_2, \dots, n_m trong đó n_i ($1 \leq n_i \leq 10^6$) – số xu trong tháp ở ván chơi thứ i .

Kết quả: Đưa ra tập tin văn bản **COINTOWER.OUT** xâu m ký tự từ tập $\{A,B\}$, ký tự thứ i là A nếu Alice thắng được và là B nếu Bob thắng.

Ví dụ:

COINTOWER . INP	COINTOWER . OUT
2 3 5 3 12 113 25717 88888	ABAAB