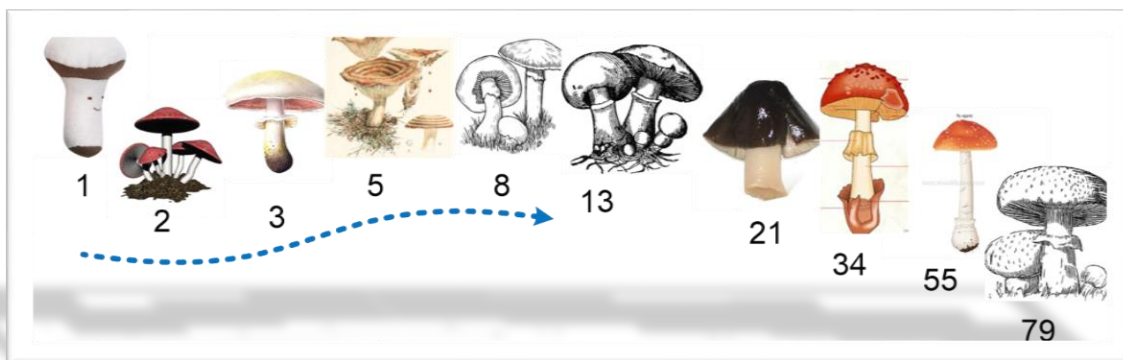


HÁI NẤM

Tên chương trình: MUSHROOM.???

Trước mặt Mario là một dãy 10 cây nấm xếp thành một hàng dài, mỗi cây nấm có một giá trị riêng, là một số nguyên không vượt quá 100. Mario không cần phải hết nấm mà chỉ cần đạt tổng giá trị càng gần 100 càng tốt và chỉ được hái liên tục các cây nấm cạnh nhau. Nếu có hai khả năng gần 100 tương đương nhau (ví dụ 98 và 102) Mario sẽ chọn phương án có giá trị lớn hơn.



Yêu cầu: Cho giá trị các cây nấm theo trình tự từ trái sang phải. Hãy xác định tổng giá trị nấm Mario hái được.

Dữ liệu: Vào từ file văn bản MUSHROOM.INP gồm 10 dòng, mỗi dòng chứa một số nguyên, dòng thứ *i* xác định giá trị cây nấm *i*.

Kết quả: Đưa ra file văn bản MUSHROOM.OUT một số nguyên – tổng giá trị nấm Mario hái được.

Ví dụ:

MUSHROOM.INP
1
2
3
5
8
13
21
34
55
79

MUSHROOM.OUT
110



BÓNG RỔ

Tên chương trình: BASKET_B.???

Steve theo dõi một trận đấu bóng rổ trong giải NBA. Trận đấu kéo dài đúng 48 phút. Sau khi kết thúc, trên màn hình điện tử hiện lên bảng thống kê cho biết lúc mỗi đội ghi điểm trong trận đấu. Thông tin mỗi dòng có dạng

k mm:ss

trong đó ***k*** – đội ghi điểm, ***k***= 1 hoặc 2, mỗi lần ghi được 1 điểm,

mm – phút, ***ss*** – giây : thời điểm ghi điểm, $0 \leq mm \leq 47$, $0 \leq ss \leq 59$.

Steve muốn tính tổng thời gian mỗi đội vươn lên dẫn trước.

Yêu cầu: Cho ***n*** ($1 \leq n \leq 100$) và các dòng thông báo. Hãy tính tổng thời gian mỗi đội vươn lên dẫn trước.

Dữ liệu: Vào từ file văn bản BASKET_B.INP:

- Dòng đầu tiên chứa số nguyên ***n***,
- Mỗi dòng trong ***n*** dòng sau chứa một thông báo theo dạng đã nêu.

Kết quả: Đưa ra file văn bản BASKET_B.OUT:

- Dòng thứ nhất: tổng thời gian đội thứ nhất dẫn đầu,
- Dòng thứ hai: tổng thời gian đội thứ hai dẫn đầu.

Thông tin trên mỗi dòng có quy cách ***mm:ss***.

Ví dụ:

BASKET_B.INP
3
1 01:10
2 21:10
2 31:30

BASKET_B.OUT
20:00
16:30



CẬP BẾN

Tên chương trình: LANDFALL.???

Steve sống ở một thị trấn nhỏ. Ở đây có một bến cảng mà năm thì mười họa mới có một tàu cập bến. Hôm nay là một ngày đáng nhớ bởi vì tất cả các tàu ra vào cảng này đều cập bến. Steve đánh dấu nó là ngày 1.

Ngày tháng trôi qua phẳng lặng. Ngày có tàu cập bến được Steve coi là thú vị và được ghi lại trên lịch. Steve nhận thấy là các tàu ghé vào cảng theo chu kỳ nhất định, mỗi tàu có một chu kỳ riêng của mình. Ví dụ, nếu chu kỳ là 3 ngày thì tàu ghé cảng vào các ngày 1, 4, 7, 10, ...

Yêu cầu: Cho n – số lần có tàu ghé cảng ($1 \leq n \leq 5000$) và các ngày có tàu cập bến. Hãy xác định số lượng tàu khác nhau ít nhất cập bến.

Dữ liệu: Vào từ file văn bản LANDFALL.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Mỗi dòng trong n dòng sau chứa một số nguyên – ngày tàu cập bến (không vượt quá 10^9).

Kết quả: Đưa ra file văn bản LANDFALL.OUT một số nguyên – số lượng tàu khác nhau ít nhất cập bến.

Ví dụ:

LANDFALL.INP
5
1
7
10
13
19

LANDFALL.OUT
2



LỰA CHỌN

Tên chương trình: CHOICE.???

Ban Giám khảo có một loạt các bài đề xuất cho cuộc thi Olympic Tin học sắp tổ chức. Độ khó của mỗi bài được đánh giá bằng một số nguyên trong phạm vi từ 1 đến n . Tuy vậy có những bài không dễ dàng đánh giá độ khó. Độ khó của một bài như vậy có thể là i hoặc $i+1$. Khi đó bài ấy có thể sử dụng như là bài ở độ khó của một trong 2 mức đã nói.

Ban Giám khảo cần tạo một đề gồm n bài, ở mỗi độ khó có đúng một bài và không có 2 bài giống nhau trong đề.

Yêu cầu: Cho số nguyên n và các số nguyên $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_{n-1}$, trong đó a_i – số bài đề xuất có độ khó i , b_i – số bài độ khó i có thể coi là có độ khó $i+1$ ($1 \leq n \leq 10^5$, $0 \leq a_i, b_j \leq 10^9$). Hãy tính số cách khác nhau tạo đề thi. Hai đề thi gọi là khác nhau nếu tồn tại ít nhất một bài có ở trong đề này và không có trong đề kia. Số cách chọn có thể rất lớn vì vậy chỉ cần đưa ra số cách chọn theo mô đun 10^9+7 .

Dữ liệu: Vào từ file văn bản CHOICE.INP:

- Dòng thứ nhất chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ,
- Dòng thứ 3 chứa các số nguyên b_1, b_2, \dots, b_{n-1} .

Kết quả: Đưa ra file văn bản CHOICE.OUT một số nguyên – số cách chọn theo mô đun 10^9+7 .

Ví dụ:

CHOICE.INP
3
3 0 1
0 1

CHOICE.OUT
3



DÃY HẤP DẪN

Tên chương trình: INTEREST.???

Dãy $2 \times k$ số nguyên gọi là hấp dẫn nếu cả tổng k phần tử đầu lẫn tổng k phần tử cuối đều không lớn hơn s cho trước.

Yêu cầu: Cho n , s và các phần tử a_1, a_2, \dots, a_n ($2 \leq n \leq 10^5$, $1 \leq s \leq 2 \times 10^9$). Các số a_i đều nguyên dương và tổng của chúng không vượt quá 2×10^9 . Với mỗi i , $i = 1 \div n$, hãy xác định độ dài dãy con thứ vị gồm các phần tử liên tiếp nhau bắt đầu từ phần tử a_i .

Dữ liệu: Vào từ file văn bản INTEREST.INP:

- Dòng đầu tiên chứa 2 số nguyên n và s ,
- Dòng thứ i trong n dòng sau chứa số nguyên a_i .

Kết quả: Đưa ra file văn bản INTEREST.OUT n số nguyên, mỗi số trên một dòng, số thứ i là độ dài dãy con thứ vị gồm các phần tử liên tiếp nhau bắt đầu từ phần tử a_i .

Ví dụ:

INTEREST.INP	
8	3
1	
1	
1	
1	
1	
1	
1	
1	
1	

INTEREST.OUT	
6	
6	
6	
4	
4	
2	
2	
0	



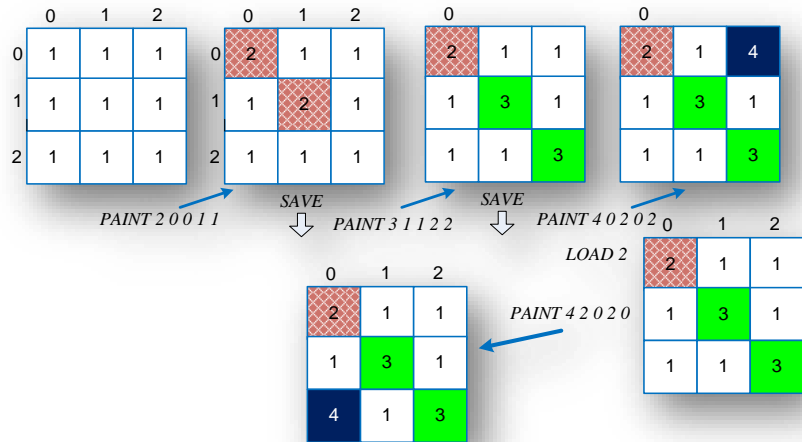
TRANH SƠN DẦU

Tên chương trình: CANVAS.???

Steve cài đặt chương trình tạo tranh sơn dầu. Chương trình hỗ trợ k màu đánh số từ 1 đến k . Tranh được vẽ trên mảnh vải kích thước $n \times n$ ô. Các hàng được đánh số từ 0 từ trên xuống dưới (tọa độ x), các cột (tọa độ y) đánh số từ 0, từ trái sang phải. Ban đầu các ô đều có màu trắng (màu 1).

Chương trình cung cấp 3 lệnh xử lý:

- **PAINT c $x1$ $y1$ $x2$ $y2$** – chọn hình chữ nhật có góc trên trái ($x1$, $y1$), góc dưới phải là ($x2$, $y2$) và tô hình chữ nhật này theo kiểu vẽ bàn cờ: tô một ô và để cách một ô với ô trên trái là màu c , các ô không tô vẫn giữ nguyên màu cũ,
- **SAVE** – lưu tranh hiện tại,
- **LOAD j** – lấy bức tranh được lưu ở lệnh **SAVE** thứ j (đánh số từ 1 trở đi) ra xử lý.



Chương trình vẽ bị nhiễm vi rút và bị hệ thống bảo vệ xóa. Nhưng may mắn, Steve vẫn còn lưu được biên bản xử lý – dãy các câu lệnh đã kích hoạt, gồm tất cả m câu lệnh.

Yêu cầu: Cho n , k , m và dãy các câu lệnh đã thực hiện ($1 \leq n \leq 1000$, $2 \leq k \leq 10^5$, $1 \leq m \leq 10^5$). Tất cả các câu lệnh đều hợp lệ. Hãy khôi phục lại bức tranh đã vẽ.

Dữ liệu: Vào từ file văn bản CANVAS.INP:

- Dòng đầu tiên chứa 3 số nguyên n , k và m ,
- Mỗi dòng trong m dòng sau chứa 1 câu lệnh trong biên bản đã lưu.

Kết quả: Đưa ra file văn bản CANVAS.OUT bức tranh dưới dạng ma trận nguyên kích thước $n \times n$.

Ví dụ:

CANVAS.INP				
3	4	7		
PAINT	2	0	0	1 1
SAVE				
PAINT	3	1	1	2 2
SAVE				
PAINT	4	0	2	0 2
LOAD	2			
PAINT	4	2	0	2 0

CANVAS.OUT				
2	1	1		
1	3	1		
4	1	3		



XÓA ĐUÔI

Tên chương trình: SUFFIX.???

Để rèn luyện kỹ năng xử lý xâu thầy giáo viết một xâu chuẩn bị trước lên bảng, lần lượt từng ký tự, từ trái sang phải. Nếu xuất hiện một phần đuôi không rỗng là xâu con trong phần bên trái còn lại của xâu thì học sinh phải kêu lên: "Xóa đuôi!". Thầy giáo sẽ xóa phần đuôi này rồi viết tiếp các ký tự còn lại vào cuối xâu đang có trên bảng.

Yêu cầu: Cho xâu ban đầu mà thầy giáo chuẩn bị. Xâu có độ dài không quá 1000 và chỉ chứa các ký tự la tinh thường. Hãy xác định xâu còn lại trên bảng.

Dữ liệu: Vào từ file văn bản SUFFIX.INP gồm một dòng chứa xâu ban đầu.

Kết quả: Đưa ra file văn bản SUFFIX.OUT xâu còn lại trên bảng.

Ví dụ:

SUFFIX.INP
abcabc

SUFFIX.OUT
abc



SỐ PHẪNG LẶNG

Tên chương trình: SMOOTH.???

Số phẳng lặng là số mà ngoại trừ 2 chữ số đầu và cuối, mỗi chữ số còn lại đều bé hơn trung bình cộng của chữ số đứng trước và sau nó, tức là với

$$x = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10 + a_0$$

có
$$a_i < \frac{a_{i+1} + a_{i-1}}{2}, i = 1 \div n-1$$

Steve say mê tìm hiểu về các số phẳng lặng và muốn biết là với n nguyên cho trước ($1 \leq n \leq 100$) có tồn tại số phẳng lặng có đúng n chữ số và nếu có thì số lớn nhất là bao nhiêu.

Yêu cầu: Cho n . Hãy xác định số phẳng lặng lớn nhất có đúng n chữ số. Nếu không tồn tại – đưa ra số -1.

Dữ liệu: Vào từ file văn bản SMOOTH.INP gồm một dòng chứa số nguyên n .

Kết quả: Đưa ra file văn bản SMOOTH.OUT giá trị xác định được.

Ví dụ:

SMOOTH.INP
3

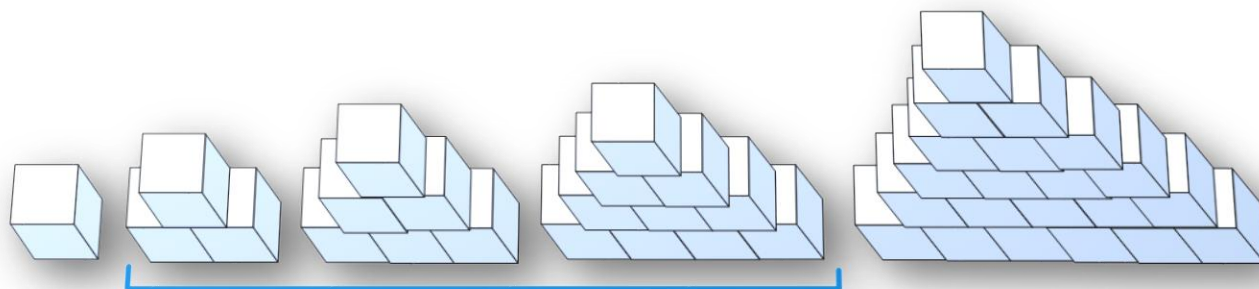
SMOOTH.OUT
989



KHỐI LẬP PHƯƠNG

Tên chương trình: CUBICS.???

Quà sinh nhật của Jimmy là một bộ khối lập phương xếp hình. Jimmy xếp thành n tháp, tháp thứ i có độ cao là a_i ($1 \leq a_i \leq 10^9$, $1 \leq n \leq 10^5$, $i = 1 \div n$).



Jimmy rất có cảm tình với số nguyên k , vì vậy dãy liên tục các tháp được coi là hài hòa nếu chúng có độ cao trung bình là k ($1 \leq k \leq 10^9$).

Yêu cầu: Cho n , k và a_i , $i = 1 \div k$. Hãy xác định dãy tháp hài hòa dài nhất, chỉ ra tháp đầu tiên và độ dài của dãy tìm được. Nếu tồn tại nhiều dãy cùng độ dài thì chỉ ra dãy tháp có vị trí đầu nhỏ nhất. Nếu không tồn tại dãy tháp thì đưa ra một số 0.

Dữ liệu: Vào từ file văn bản CUBICS.INP:

- Dòng đầu tiên chứa 2 số nguyên n và k ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản CUBICS.OUT trên một dòng 2 số nguyên: độ dài của dãy tìm được và số thứ tự của tháp đầu tiên hoặc một số 0 nếu không tồn tại dãy.

Ví dụ:

CUBICS.INP					
5	3				
1	2	3	4	6	

CUBICS.OUT	
3	2



VẦN ĐIỆU

Tên chương trình: RHYTME.???

Giáo sư Johson dự định tuyên truyền, phổ biến rộng rãi thơ song tấu. Đó loại thơ mà mỗi bài chỉ có 2 câu có từ cuối vần với nhau và mỗi câu có đúng s nguyên âm.

Hai từ gọi là vần với nhau nếu có hậu tố (**suffix**) của mỗi từ tính từ nguyên âm cuối cùng giống nhau, ví dụ “**cat**” và “**chat**”, “**processing**” và “**interesting**”, “**castle**” và “**table**”,. . . .

Để tạo ví dụ minh họa, giáo sư chuẩn bị danh sách n từ khác nhau, mỗi từ có độ dài không quá 30, chỉ bao gồm các ký tự la tinh thường. Trong bảng chữ cái la tinh, nguyên âm là các ký tự trong tập $\{a, e, i, o, u, y\}$.

Yêu cầu: Cho n , s và các từ trong danh sách ($0 \leq n \leq 10^5$, $1 \leq s \leq 10^5$). Hãy đưa ra một bài thơ song tấu từ các từ đã cho. Nếu không tồn tại bài thơ nào thì đưa ra thông báo “**Impossible**”.

Dữ liệu: Vào từ file văn bản RHYTME.INP:

- Dòng đầu tiên chứa 2 số nguyên n và s ,
- Mỗi dòng trong n dòng sau chứa một từ.

Kết quả: Đưa ra file văn bản RHYTME.OUT bài thơ song tấu xây dựng được, mỗi dòng chứa một câu hoặc thông báo “**Impossible**”.

Ví dụ:

RHYTME.INP
7 6
my
food
is
tasty
cat
looks
happy

RHYTME.OUT
cat food is tasty
my cat looks happy



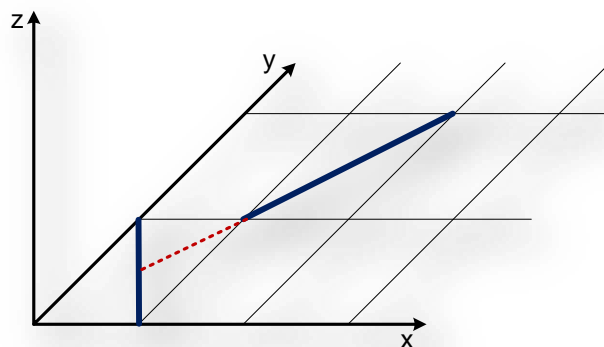
ARDENIA

Tên chương trình: ARDENIA.???

Chào mừng bạn đã đến Ardenia, một đất nước thần thoại với đầy các chuyện phiêu lưu và mạo hiểm, xứ sở của các rồng và giới, của các thầy phù thủy và chiêm tinh, xứ sở của các bài toán đố. Rất nhiều bài toán đố. Giải toán đố là niềm đam mê của người dân ở đây, là điều liên kết mọi người.

Tháng này, bài toán ra cho mọi người là tìm khoảng cách giữa hai đoạn thẳng trong không gian 3 chiều. Khoảng cách giữa 2 đoạn thẳng là độ dài ngắn nhất nối 2 điểm, mỗi điểm trên một đoạn thẳng đã cho. Thực ra, bài toán này có xuất xứ từ một vấn đề thực tế, nhưng chẳng ai trên Ardenia quan tâm đến vấn đề đó và dĩ nhiên bạn lại càng không cần lưu ý đến nó.

Yêu cầu: Cho các tọa độ nguyên $(x1, y1, z1)$, $(x2, y2, z2)$ – 2 điểm đầu và cuối xác định đoạn thẳng thứ nhất và $(u1, v1, w1)$, $(u2, v2, w2)$ – 2 điểm đầu và cuối xác định đoạn thẳng thứ hai. Tất cả các tọa độ có giá trị tuyệt đối không vượt quá 20. Hãy xác định 2 số nguyên tố cùng nhau k và m sao cho k/m là bình phương khoảng cách tìm được. Nếu khoảng cách bằng 0 thì đưa ra kết quả với $k = 0$ và $m = 1$. Các tọa độ điểm đầu và cuối mỗi đoạn thẳng không trùng nhau.



Dữ liệu: Vào từ file văn bản ARDENIA.INP:

- Dòng đầu tiên chứa số nguyên n – số lượng tests ($1 \leq n \leq 10^5$),
- Mỗi test được cho trên một nhóm 2 dòng:
 - Dòng đầu tiên trong nhóm chứa 6 số nguyên $x1, y1, z1, x2, y2, z2$,
 - Dòng thứ 2 trong nhóm chứa 6 số nguyên $u1, v1, w1, u2, v2, w2$.

Kết quả: Đưa ra file văn bản ARDENIA.OUT, kết quả mỗi test đưa ra trên một dòng chứa 2 số nguyên k và m tương ứng.

Ví dụ:

ARDENIA.INP						
2						
0	0	0	1	1	1	
1	1	1	2	2	2	
1	0	0	0	1	0	
1	1	0	2	2	0	

ARDENIA.OUT	
0	1
1	2



KHOẢNG CÁCH

Tên chương trình: DISTANCE.???

Hai con thú Lửa (F) và Băng (I) là báu vật tiền sử còn lại của Ardenia, mỗi con sống ở một hang riêng tại các điểm có tọa độ $(0, -10^9)$ và $(0, 10^9)$. Các nhà pháp thuật đã tạo ra một loạt các hàng rào vô hình dưới dạng những đường thẳng vô tận, các con thú không vượt qua được những hàng rào này. Các hàng rào có thể cắt nhau, nhưng không đi ngang qua hang. Tuy vậy, thời gian trôi đi và một vài hàng rào bị mất hiệu lực. Nếu 2 con thú gặp nhau thì sẽ xảy ra một thảm họa khủng khiếp cho toàn xứ Ardenia.

Quốc vương Ardenia quan tâm đến sự an toàn của vương quốc và cử tể tướng đi thành lập bản đồ các hàng rào còn tồn tại, tính khoảng cách mà hai con thú có thể tới gần nhau nhất. Vương quốc có thể coi như một mặt phẳng vô hạn. Mỗi hàng rào xác định bằng một phương trình đường thẳng $ax+by+c=0$ ($-10^9 \leq a, b, c \leq 10^9$, $b \neq 0$, a, b, c – nguyên).

Yêu cầu: Cho n – số hàng rào còn tồn tại ($1 \leq n \leq 2 \times 10^5$) và các hệ số a, b, c của mỗi hàng rào. Hãy xác định khoảng cách gần nhất có thể giữa hai con thú với độ chính xác 10^{-6} .

Dữ liệu: Vào từ file văn bản DISTANCE.INP:

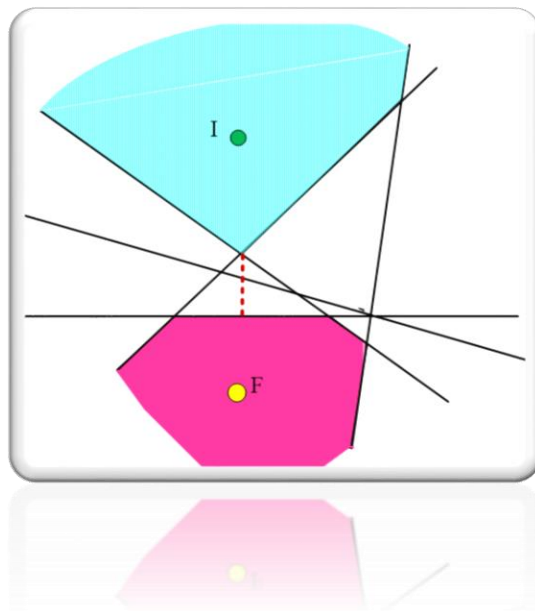
- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 20$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa số nguyên n ,
 - Mỗi dòng trong n dòng tiếp theo chứa 3 số nguyên a, b, c xác định một hàng rào.

Kết quả: Đưa ra file văn bản DISTANCE.OUT, kết quả mỗi test đưa ra trên một dòng với độ chính xác 10^{-6} .

Ví dụ:

DISTANCE.INP			
2			
5			
1	-1	0	
1	1	0	
0	1	-6	
0	1	-10	
0	1	10	
4			
1	1	10	
2	6	7	
-1	2	10	
0	1	12	

DISTANCE.OUT	
400.000000	
93.250000	



CÂU THẦN CHÚ

Tên chương trình: SPELLS.???

Việc lập đi lập lại các âm điệu trong các câu thần chú là điều ít được nghiên cứu nhất hiện nay. Thực tế, khi nghe, người ta khó phân biệt câu “**abrahellehhelleh**” (dùng trong các trận đánh hay ở trường dạy phù thủy với câu “**rachelhellabracadabra**” mà các nhà phù thủy trình độ thấp mắng con mèo.

Cuối cùng thì các công trình nghiên cứu của trường Đại học Ấn danh cũng đã cho biết cách đánh giá sức mạnh của các câu thần chú. Điều này liên quan tới khả năng đọc ngược và xuôi một từ của người niệm thần chú (quả thật, với khả năng này, một số người đã bị thiêu sống vì bị coi là có quyền lực liên hệ với quỷ dữ). Sức mạnh của câu thần chú được xác định bởi độ dài xâu con dài nhất có dạng ww^rw^r trong câu thần chú. Ở đây w là một dãy các ký tự liên tiếp, còn w^r là dãy ký tự đó viết theo trình tự ngược lại. Nếu không tồn tại xâu con dạng này, câu thần chú có sức mạnh bằng 0.

Ví dụ, câu thần chú “**abrahellehhelleh**” có sức mạnh 12 vì chứa xâu con “**hellehhelleh**”, còn câu “**rachelhellabracadabra**” có sức mạnh bằng 0. Lưu ý là sức mạnh câu thần chú luôn chia hết cho 4.

Yêu cầu: Cho câu thần chú có độ dài không quá 3×10^5 chỉ chứa các ký tự la tinh hoa và thường. Hãy xác định sức mạnh câu thần chú.

Dữ liệu: Vào từ file văn bản SPELLS.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 40$),
- Mỗi dòng trong t dòng sau chứa một câu thần chú.

Kết quả: Đưa ra file văn bản SPELLS.OUT các số nguyên – sức mạnh của các câu thần chú, mỗi số trên một dòng.

Ví dụ:

SPELLS.INP
2
abrahellehhelleh
rachelhellabracadabra

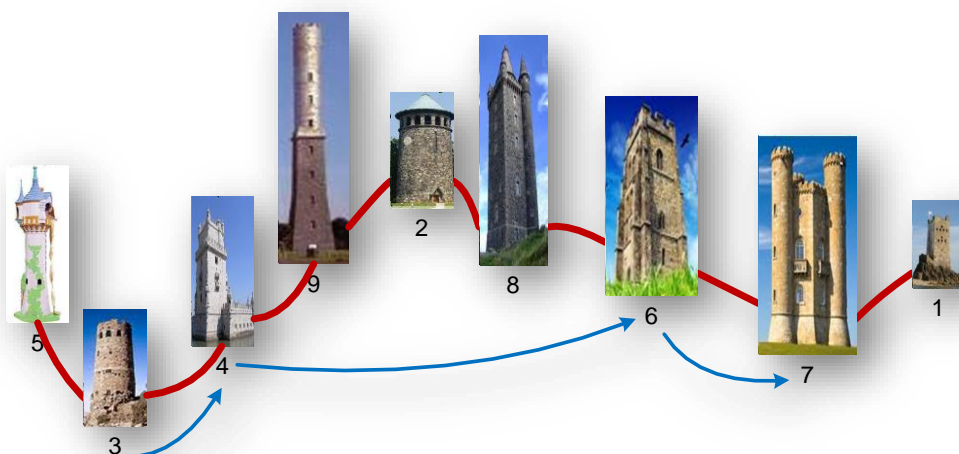
SPELLS.OUT
12
0



TUYẾN PHÒNG VỆ

Tên chương trình: DEFENSE.???

Sau cuộc chiến tranh gần nhất, đất nước bị tàn phá nặng nề. Quốc vương xứ Ardenia quyết định củng cố tuyến phòng thủ thủ đô. Tuyến phòng thủ bao gồm dãy các pháo đài nằm trên đường nối thủ đô với khu rừng ở phía bắc, bao gồm n pháo đài đánh số từ 1 đến n kể từ pháo đài gần thủ đô nhất, pháo đài thứ i có độ cao h_i ($1 \leq h_i \leq 10^9$, $1 \leq n \leq 2 \times 10^5$, $i = 1 \div n$). Quốc vương yêu cầu các quân sư đánh giá khả năng phòng thủ. Các quân sư trình bày khá nhiều và lâu, nhưng nhà vua chỉ nắm được là chất lượng tuyến phòng thủ phụ thuộc vào dãy con các pháo đài có độ cao tăng dần tính từ thủ đô (do đặc trưng cách vây đánh của kẻ địch).



Ví dụ, với $n = 9$ và độ cao các pháo đài là (5, 3, 4, 9, 2, 8, 6, 7, 1) thì dãy các pháo đài với độ cao (3, 4, 6, 7) sẽ quyết định chất lượng của toàn tuyến phòng thủ.

Yêu cầu: Cho n và các h_i . Hãy xác định số lượng pháo đài quyết định chất lượng của toàn tuyến phòng thủ.

Dữ liệu: Vào từ file văn bản DEFENSE.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 25$),
- Mỗi test cho trên 2 dòng:
 - Dòng thứ nhất chứa số nguyên n ,
 - Dòng thứ 2 chứa n số nguyên h_1, h_2, \dots, h_n .

Kết quả: Đưa ra file văn bản DEFENSE.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên – số lượng pháo đài quyết định chất lượng của toàn tuyến phòng thủ.

Ví dụ:

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

DEFENSE.OUT	
4	
6	



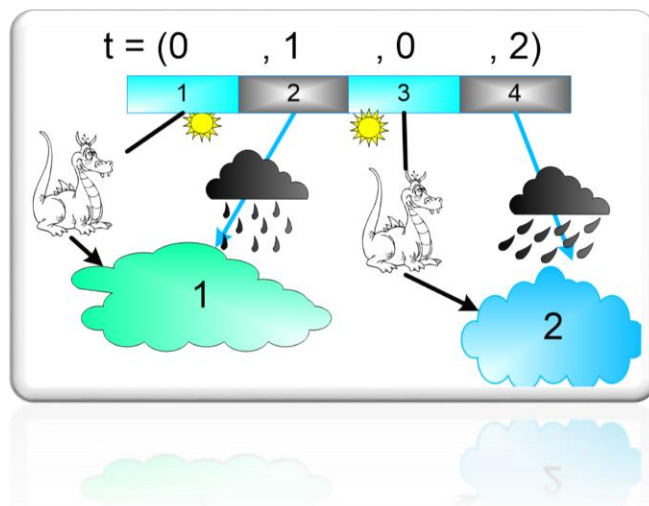
RỒNG HÚT NƯỚC

Tên chương trình: DRAGON.???

Bao quanh thủ đô Ardenia là n hồ lớn, đánh số từ 1 đến n . Ban đầu các hồ đều đầy nước. Theo dự báo thời tiết những ngày tới sẽ có các trận mưa lớn, mỗi trận mưa diễn ra trên vùng của một hồ. Nếu hồ cạn, mưa sẽ làm hồ đầy nước, nếu hồ đang đầy nước, mưa sẽ làm nước tràn bờ gây ngập lụt ở thủ đô. May mắn là người dân thủ đô có một con rồng. Vào ngày không mưa người ta có thể dắt rồng một hồ nào đó và trong ngày rồng có thể uống cạn hồ nước này.

Cơ quan dự báo thời tiết đưa ra dự báo cho m ngày dưới dạng các số t_1, t_2, \dots, t_m , trong đó $t_i \in [0, n]$. Nếu $t_i \in [1, n]$ có nghĩa là sẽ có mưa trên hồ t_i vào ngày i , nếu $t_i = 0$ là ngày i không có mưa.

Yêu cầu: Cho n, m và các t_i ($1 \leq n, m \leq 10^6, i = 1 \div m$). Hãy xác định xem có tồn tại lịch cho rồng hút nước tránh lụt được hay không và đưa ra câu trả lời “**YES**” hoặc “**NO**”. Nếu có lịch hút nước thì đưa ra ở dòng tiếp theo các số nguyên l_1, l_2, \dots, l_p , trong đó p là số ngày không mưa, $l_j \in [0, n]$. Nếu $l_j \neq 0$ có nghĩa là cần hút nước ở hồ l_j trong ngày không mưa thứ j .



Dữ liệu: Vào từ file văn bản DRAGON.INP:

- Dòng đầu tiên chứa số nguyên q – số lượng tests ($1 \leq q \leq 40$),
- Mỗi test cho trên 2 dòng:
 - Dòng thứ nhất chứa các số nguyên n và m ,
 - Dòng thứ 2 chứa m số nguyên t_1, t_2, \dots, t_m .

Kết quả: Đưa ra file văn bản DRAGON.OUT, kết quả mỗi test đưa ra trên một hoặc 2 dòng:

- Dòng thứ nhất chứa thông báo “**YES**” hoặc “**NO**”,
- Nếu kết quả là “**YES**” thì ở dòng thứ 2 đưa ra các số nguyên l_1, l_2, \dots, l_p .

Ví dụ:

DRAGON.INP				
4				
2	4			
0	0	1	1	
2	4			
0	1	0	2	
2	3			
0	1	2		
2	4			
0	0	0	1	

DRAGON.OUT		
NO		
YES		
1	2	
NO		
YES		
0	1	0



d15 CERC 10 E

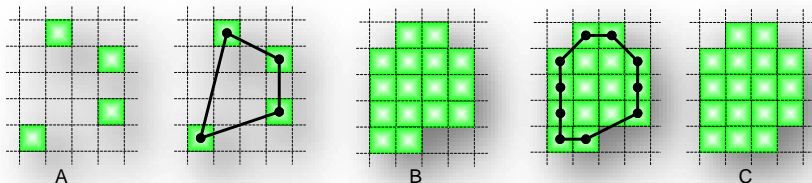
CÁNH ĐỒNG

Tên chương trình: FIELDS.???

Làm chủ nông trang ở Ardenia không phải là dễ dàng. Chưa nói đến thái độ lạnh lùng và hằn học đối với các cánh đồng cỏ chăn nuôi gia súc quốc vương Ardenia không quan tâm đến phát triển nông nghiệp. Ông muốn người dân của mình đi lấn chiếm đất láng giềng và nuôi trồng gì ở đó cũng được, vì vậy đã bản cùng hóa nông dân đến mức giới hạn. Ngay đến việc tương chừng như đơn giản – mua vài lô đất nhỏ cũng là vấn đề lớn với người dân.

Toàn bài thửa nuôi gia súc có hình chữ nhật được chia thành các lô hình vuông đơn vị. Ban đầu người ta mua một số lô nào đó. Bãi chăn gia súc sẽ được xác định như sau:

- Cắm cọc vào tâm các ô thuộc sở hữu của mình,
- Chăng giầy tạo thành một đa giác lồi diện tích nhỏ nhất chứa tất cả các cọc (trên biên hoặc bên trong),
- Nếu đường dây cắt một lô nào đó với diện tích lớn hơn 0 bên trong đa giác thì cắm cọc sở hữu luôn ô đó,
- Làm lại việc chằng giầy và sở hữu lô mới cho đến khi không xuất thêm lô nào nữa.



Có nhiều cách mua các tập con các lô ban đầu để kết quả chằng giầy mở rộng cùng ra một kết quả như nhau. Những cách mua như vậy gọi là tương đương. Hai cách mua gọi là khác nhau nếu tồn tại ít nhất một lô có ở cách mua này nhưng không có ở cách mua kia.

Yêu cầu: Cho n – số lô mua ban đầu và tọa độ nguyên (x_i, y_i) của các ô được mua ($1 \leq n \leq 10^6$, $x_i, y_i \in [-10^9, 10^9]$, $i = 1 \div n$). Hãy xác định số cách mua tương đương và đưa ra theo mô đun 10^9+7 .

Dữ liệu: Vào từ file văn bản FIELDS.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 50$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng thứ nhất trong nhóm chứa số nguyên n ,
 - Dòng thứ i trong n dòng sau chứa 2 số nguyên x_i và y_i .

Kết quả: Đưa ra file văn bản FIELDS.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên.

Ví dụ:

FIELDS.INP
2
4
0 0
0 1
0 2
0 3
5
0 0
-1 0
1 0
0 -1
0 1

FIELDS.OUT
4
2



d16 CERC 10 F

TRÒ CHƠI 5×5

Tên chương trình: GAME25.???

Trò chơi đặt sỏi trên bảng 5×5 là hình thức giải trí giết thời gian phổ biến thứ 3 trên Ardenia (chỉ đứng sau trò giải toán đồ và nói xấu sau lưng láng giềng). Hai người lần lượt đặt sỏi của mình trên bàn cờ kích thước 5×5 ô vuông. Chúng ta không quan tâm đến luật đi, chỉ cần biết rằng mục tiêu của mỗi người chơi là khi kết thúc (bàn cờ đã đặt kín sỏi, mỗi ô một viên), mỗi quân của mình đều thuộc một dãy 3 quân của mình liên tiếp theo đường ngang, dọc hoặc chéo. Nếu cả hai người đều cùng đạt được mục đích hay cùng không đạt mục đích thì ván cờ coi như hòa. Nếu chỉ một người đạt mục đích thì người đó thắng.

Hai anh em Arthur và Bruce không thích trò chơi này lắm, nhưng mẹ cứ bắt phải chơi. Arthur đặt các viên đá có ký hiệu **A**, còn Bruce – các viên có ký hiệu **B**.

Đã 2 năm trôi qua bỗng nhiên hai người lại nhìn thấy ảnh chụp ván cờ họ đã chơi và muốn tính lại xem người đi viên sỏi **A** thắng hay người đi viên sỏi **B** thắng hoặc có thể đây là ván hòa.



Yêu cầu: Cho 5 xâu ký tự độ dài 5 từ tập $\{A, B\}$ mô tả bàn cờ. Hãy đưa ra kết quả dưới dạng thông báo “**A wins**” hoặc “**B wins**” hoặc “**draw**”.

Dữ liệu: Vào từ file văn bản GAME25.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 10^5$),
- Mỗi test cho trên một nhóm 5 dòng mô tả bàn cờ, mỗi dòng một xâu độ dài 5.

Kết quả: Đưa ra file văn bản GAME25.OUT các xâu thông báo kết quả, mỗi xâu trên một dòng.

Ví dụ:

GAME25.INP
2
AABBA
BAAAB
AAABA
ABAAB
BAAAB
AAAAA
AAAAA
BAAAA
ABAAA
AABAA

GAME25.OUT
A wins
draw



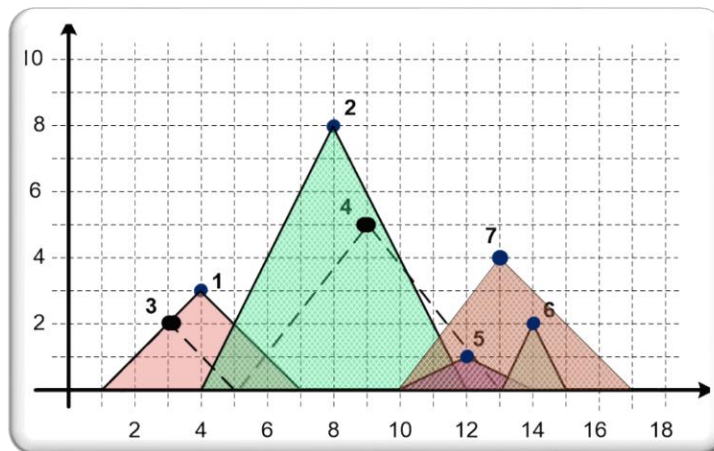
TREO MŨ

Tên chương trình: HATS.???

Ai cũng biết rằng phù thủy thì phải đội mũ, nhưng ít ai biết là khi ngủ họ không đội mũ. Ở trường Đại học Ẩn danh có một bức tường lớn để để mọi người treo mũ trước khi đi ngủ. Mũ mà mọi người đội có hai loại: Rộng (W) và hẹp (N).

Không phải mọi người đi ngủ cùng một lúc vì nhà trường chỉ có một phòng tắm và một cái khăn mặt. Ai làm vệ sinh cá nhân xong thì vào treo mũ. Đến lượt mình người thứ i lấy một cái đỉnh do nhà trường cấp, đóng vào vị trí có tọa độ nguyên (x_i, y_i) , treo mũ lên đó ($y_i > 0$ và là khoảng cách từ đỉnh đến sàn). Mũ cũng biết biến hình. Khi được treo nó sẽ biến thành hình dạng tam giác cân có đỉnh lầy nơi đóng đinh và đáy ở chân tường. Mũ rộng có đáy dài gấp đôi đường cao, mũ loại hẹp có đáy đúng bằng đường cao.

Nếu ai đó đóng đinh vào vị trí đã có đỉnh hoặc xuyên vào mũ đã treo (kể cả trên biên) đỉnh sẽ bị rơi và người đó lập tức bị đuổi ra khỏi trường. Nếu đỉnh bị mũ che (kể cả trên biên) sẽ không phát sáng, nhưng đỉnh còn lại – phát sáng trong đêm. Trưởng phòng Quản trị của trường mỗi đêm ghi lại số lượng đỉnh phát sáng sau khi có một người vào treo mũ (có lẽ ông ta không có điều gì khác phải làm).



Yêu cầu: Cho n – số lượng phù thủy ($1 \leq n \leq 10^5$) và các tọa độ x_i, y_i ($-10^9 \leq x_i \leq 10^9, 1 \leq y_i \leq 10^9, i = 1 \div n$). Hãy xác định số đỉnh phát sáng sau mỗi lần có người treo mũ. Nếu một người đóng đinh sai, bị đuổi thì đưa ra thông báo “FAIL”.

Dữ liệu: Vào từ file văn bản HATS.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 30$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng thứ nhất trong nhóm chứa số nguyên n ,
 - Dòng thứ i trong n dòng sau chứa 2 số nguyên x_i và y_i .

Kết quả: Đưa ra file văn bản HATS.OUT, kết quả mỗi test đưa ra trên n dòng mỗi dòng một số nguyên – số đỉnh phát sáng.

Ví dụ:

HATS.INP		
2		
3		
0	1	W
0	2	N
0	1	W
7		
4	3	W
8	8	N
3	2	W
9	5	W
12	1	W
14	2	N
13	4	W

HATS.OUT	
1	
1	
FAIL	
1	
2	
FAIL	
FAIL	
3	
4	
3	



TRÊU CHỌC

Tên chương trình: TEASE.???

Trêu chọc bạn bè và hàng xóm láng giềng cũng là một môn thể thao ở xứ Ardenia. Mỗi môn thể thao đều có luật của nó. Các từ dùng để trêu chọc chỉ được xây dựng từ 4 nguyên âm **a**, **e**, **i** và **o**, hơn thế nữa, không phải mọi từ mà chỉ những từ theo luật sau:

- **ae** và **io** là hai từ ngắn nhất,
- Nếu **w₁** và **w₂** là 2 từ trêu chọc thì các từ **w₁w₂**, **aw₁e**, **iw₁o** cũng là các từ trêu chọc,
- Chỉ có các từ theo 2 quy tắc trên có thể sử dụng.

Khi ai đó trêu chọc bạn, bạn không được phép trả lời tùy tiện. Ví dụ, nếu người ta nói “**aaeeio**” (*Cậu xử sự như một tay vắt sữa bò*), không thể trả lời “**aeio**” (*Cậu ngu quá*). Quy tắc trả lời Fair play là nói từ có cùng độ dài và thứ tự từ điển đứng ngay sau từ đối phương nói. Ví dụ như với câu nói trên, bạn phải trả lời là “**aaeioe**” (*Nhìn ở góc độ nào cũng đều thấy cậu giống một con bò*). Tồn tại lời trêu chọc tối thượng, đối phương không có câu trả lời. Ví dụ, câu trêu chọc tối thượng độ dài 8 là “**ioioioio**” (*Mẹ cậu là một con chuột đồng, còn bố là quả cây com cháy*).

Yêu cầu: Cho chuỗi **S** độ dài không vượt quá 10^6 chỉ chứa các ký tự trong tập {**a**, **e**, **i**, **o**}. Hãy xác định đó có phải là câu trêu chọc hay không. Nếu không phải – đưa ra thông báo “**INVALID**”, trong trường hợp ngược lại – đưa ra câu trả lời theo luật chơi. Nếu đó là câu tối thượng thì đưa ra thông báo “**ULTIMATE**”.

Dữ liệu: Vào từ file văn bản TEASE.INP:

- Dòng đầu tiên chứa số nguyên **t** – số lượng tests ($1 \leq t \leq 2\,000$),
- Mỗi dòng trong **t** dòng tiếp theo chứa một chuỗi **S**.

Kết quả: Đưa ra file văn bản TEASE.OUT, mỗi chuỗi kết quả đưa ra trên một dòng.

Ví dụ:

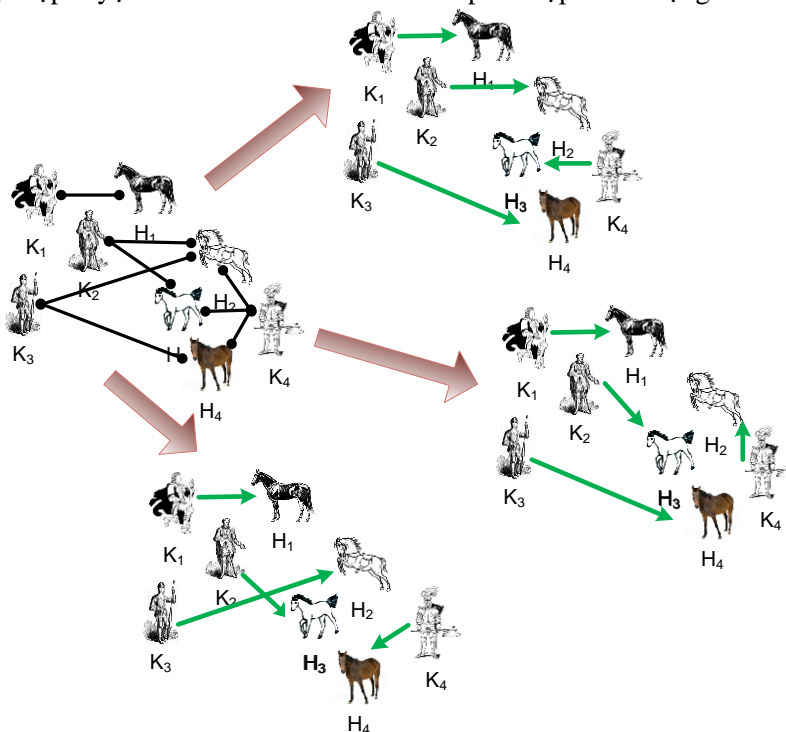
TEASE.INP	TEASE.OUT
3	INVALID
eaeeio	aaeioe
aaeeio	ULTIMATE
ioioioioio	



CÔNG BẰNG

Tên chương trình: JUSTICE.???

Xứ Ardenia lại có chiến tranh. Đội quân thiện chiến của Ardenia có 200 hiệp sỹ và 200 ngựa chiến. Để chiến đấu, người ta chỉ cần tung ra chiến trường một đơn vị gồm k kỵ sỹ và k ngựa (bộ phận còn lại vẫn ở trong doanh trại). Người và ngựa cần được tập luyện với nhau để hiểu nhau và phối hợp hành động có hiệu quả. Hiệp sỹ ra trận với ngựa mà mình đã tập luyện. Với kinh nghiệm trận mạc dày dạn họ vẫn giữ được cho mình và ngựa không bị tổn thương sau mỗi trận đánh. Mỗi trận đánh đơn vị xuất phát với một tổ hợp người và ngựa khác nhau, tức là có ít nhất một kỵ sỹ không cưỡi con ngựa mình đã dùng ở trận trước. Ví dụ với $k = 4$, hiệp sỹ I làm tập luyện với ngựa 1, hiệp sỹ II tập luyện với các ngựa 2 và 3, hiệp sỹ III tập luyện với ngựa 2 và 4, hiệp sỹ IV tập luyện với các ngựa 2, 3, 4. Khi đó có thể cho đơn vị này xuất kích 3 lần với cách sử dụng ngựa như hình đã nêu.



Các nhà chiêm tinh học dự báo sẽ có n trận đánh. Vậy phải chọn một đơn vị bao nhiêu kỵ sỹ và lên sơ đồ tập luyện ra sao để có đúng n đội hình khác nhau không hơn không kém.

Yêu cầu: Cho số nguyên n ($1 \leq n \leq 10^6$). Hãy xác định k và chỉ ra sơ đồ tập luyện. Với mỗi kỵ sỹ sơ đồ tập luyện là xâu độ dài k các ký tự $\{0, 1\}$, ký tự thứ j bằng 1 có nghĩa là kỵ sỹ sẽ tập luyện với chú ngựa thứ j .

Dữ liệu: Vào từ file văn bản JUSTICE.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi dòng trong t dòng tiếp theo chứa một số nguyên n .

Kết quả: Đưa ra file văn bản JUSTICE.OUT, kết quả mỗi test đưa ra trên một nhóm dòng, dòng đầu tiên trong nhóm là số nguyên k , mỗi dòng trong k dòng sau chứa xâu độ dài k xác định sơ đồ tập luyện.

Ví dụ:

JUSTICE.INP
3
1
3
4

JUSTICE.OUT
1
1
4
1000
0110
0101
0111
4
1100
1100
0011
0011

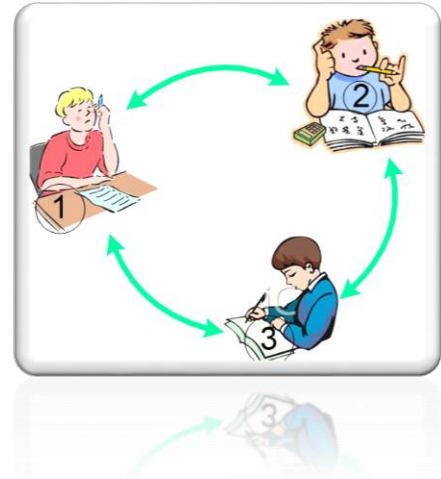


d20 CERC 10 J

ĐỘI TUYỂN

Tên chương trình: TEAMS.???

Hàng năm nhà trường thành lập các đội tuyển đi thi đồng đội về Tin học, mỗi đội có 3 người. Năm nay có n học sinh qua được vòng loại. Các học sinh này được đánh số từ 1 đến n ($1 \leq n \leq 10^5$). Nhà trường muốn thành lập được càng nhiều đội càng tốt. Tuy vậy vấn đề này cũng phải là đơn giản. Muốn đạt kết quả cao thì các thành viên trong đội phải làm việc ăn ý với nhau. Mỗi học sinh chỉ muốn ở cùng một đội với một số người bạn của mình. May mắn là quan hệ này có tính 2 chiều: **A** đồng ý làm việc với **B** thì **B** cũng đồng ý làm việc với **A**.



Yêu cầu: Cho n và m quan hệ dạng (i, j) , không có quan hệ nào trùng nhau và nếu đã có quan hệ (i, j) thì không có trong dữ liệu quan hệ (j, i) . Hãy xác định có thể thành lập nhiều nhất bao nhiêu đội tuyển.

Dữ liệu: Vào từ file văn bản TEAMS.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 20$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa 2 số nguyên n và m ,
 - Mỗi dòng trong m dòng tiếp theo chứa 2 số nguyên i và j .

Kết quả: Đưa ra file văn bản TEAMS.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên.

Ví dụ:

TEAMS.INP
2
3 3
1 2
2 3
3 1
4 4
1 2
2 3
3 4
4 1

TEAMS.OUT
1
0

CHIA CÔNG BẰNG

Tên chương trình: FAIR.???

Hôm nay là ngày sinh nhật một người bạn. Bạn và anh em trong nhóm, tất cả là n người (kể cả bạn) quyết định mua tặng một đĩa StarCraft II. Ai mà chẳng mơ ước có chương trình trò chơi này trong máy.

Bạn đề xuất phân bổ tiền mua một cách công bằng. Mỗi người sẽ đóng góp một số tiền là một số nguyên theo đơn vị xu (cent). Khả năng tài chính của mỗi người khác nhau, nhưng không ai phải đóng góp quá số tiền mà mình có.



Mỗi người ghi ra giấy số tiền mình có. Người thứ i có số tiền là a_i . Bạn phân chia phần đóng góp cho mỗi người sao cho chênh lệch với $1/n$ giá đĩa trò chơi là nhỏ nhất. Nếu phần chia bị lẻ thì phần lẻ sẽ chia đều cho những người còn nhiều tiền hơn. Nếu có nhiều người cùng có nhiều tiền như nhau và không cần tất cả phải góp thêm thì chọn người đóng góp theo thứ tự ghi trong danh sách từ trên xuống dưới.

Yêu cầu: Cho các số nguyên p , n , trong đó p là giá của chiếc đĩa trò chơi, các giá trị a_i ($1 \leq p$, $a_i \leq 10^6$, $2 \leq n \leq 100$, $i = 1 \div n$). Hãy xác định số tiền mỗi người phải góp. Nếu giá của đĩa vượt quá khả năng mua của toàn nhóm thì đưa ra thông báo “**IMPOSSIBLE**”.

Dữ liệu: Vào từ file văn bản FAIR.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi test cho trên 2 dòng:
 - Dòng đầu tiên trong nhóm chứa 2 số nguyên p và n ,
 - Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản FAIR.OUT, kết quả mỗi test đưa ra trên một dòng chứa n số nguyên – phần đóng góp của mỗi người hoặc thông báo “**IMPOSSIBLE**”.

Ví dụ:

FAIR.INP				
3				
20	4			
10	10	4	4	
7	3			
1	1	4		
34	5			
9	8	9	9	4

FAIR.OUT				
6	6	4	4	
IMPOSSIBLE				
8	7	8	7	4



CHIA BÁNH

Tên chương trình: GOODIES.???

Petra và Jan được tặng một hộp bánh, trong đó có n chiếc bánh khác nhau. Hai bạn quyết định chia nhau. Nhưng chia như thế nào cho công bằng không phải là chuyện dễ vì mỗi người có một sở thích riêng vì vậy cùng một chiếc bánh nhưng mỗi người có thể có một cách đánh giá riêng. Chiếc bánh thứ i theo cách nhìn của Petra có giá trị là p_i , còn theo Jan – giá trị là j_i .

Hai bạn quyết định lần lượt chọn, mỗi lần đến lượt mình sẽ lấy một trong số các chiếc bánh còn lại trong hộp. Ai chọn trước sẽ được quyết định bằng cách tung đồng xu.

Mỗi bạn có một chiến lược chọn riêng của mình. Khi đến lượt mình Petra luôn chọn chiếc bánh có giá trị lớn nhất (theo cách nhìn của Petra). Nếu có nhiều bánh có cùng giá trị lớn nhất Petra cẩn thận chọn chiếc bánh mà Jan ít thích nhất, tức là có giá trị nhỏ nhất đối với Jan (hai người rất thân nhau nên hiểu cách đánh giá của nhau).

Chiến lược của Jan là làm sao có tổng giá trị các bánh được chọn là lớn nhất. Khi có nhiều khả năng cùng cho giá trị lớn nhất thì Jan chọn chiếc bánh sao cho Petra cũng có tổng giá trị cuối cùng là lớn nhất có thể.

Yêu cầu: Cho n , kết quả tung xu ban đầu (“**Petra**” hoặc “**Jan**”), các giá trị p_i và j_i ($1 \leq n \leq 1\,000$, $0 \leq p_i, j_i \leq 1\,000$, $i = 1 \div n$). Hãy cho biết tổng giá trị bánh của mỗi người sau khi chia xong (theo cách đánh giá riêng tương ứng).

Dữ liệu: Vào từ file văn bản GOODIES.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa số nguyên n ,
 - Dòng thứ 2 chứa tên người chọn trước,
 - Mỗi dòng trong n dòng tiếp theo chứa 2 số nguyên p_i và j_i .

Kết quả: Đưa ra file văn bản GOODIES.OUT, kết quả mỗi test đưa ra trên một dòng gồm 2 số nguyên – tổng giá trị bánh của Petra và tổng giá trị bánh của Jan.

Ví dụ:

GOODIES.INP
3
4
Petra
100 80
70 80
50 80
30 50
4
Petra
10 1
1 10
6 6
4 4
7
Jan
4 1
3 1
2 1
1 1
1 2
1 3
1 4

GOODIES.OUT
170 130
14 16
9 10

ĐIỂM CAO

Tên chương trình: SCORE.???

Bạn chơi trò chơi video bằng cần chơi (Joystick) và đạt được điểm cao nhất. Hệ thống đề nghị bạn ghi tên mình vào danh sách những người đạt điểm cao. Sau một vài thao tác chuẩn bị trên màn hình hiện lên trường tên với độ dài đúng bằng độ dài tên bạn định đưa vào, tất cả các trường đều được điền ký tự “A”. Tất cả mọi thao tác ghi tên đều thực hiện qua cần chơi. Ký tự chọn xử lý là ký tự trái nhất của trường.

Mỗi lần gạt cần theo chiều tiến thì ký tự được chọn sẽ chuyển thành ký tự tiếp theo trong bảng chữ cái la tinh, còn nếu gạt theo chiều lùi, ký tự được chọn sẽ chuyển thành ký tự trước đó trong bảng chữ cái la tinh. Việc chuyển ký tự được thực hiện vòng tròn, tức là sau “Z” là “A” và trước “A” là ‘Z’.

Việc chuyển sang trường khác được thực hiện bằng cách gạt cần sang trái hoặc phải và việc chuyển này cũng được thực hiện vòng tròn trên trường tên.

Dĩ nhiên, bạn muốn đưa tên của mình vào một cách nhanh nhất và muốn biết số thao tác tối thiểu cần phải thực hiện là bao nhiêu.

Yêu cầu: Cho chuỗi s khác rỗng chỉ chứa các ký tự la tinh in hoa xác định tên cần đưa vào và có độ dài không quá 1 000. Hãy xác định số thao tác tối thiểu cần phải thực hiện.

Dữ liệu: Vào từ file văn bản SCORE.INP:

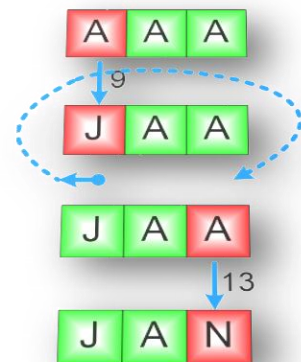
- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi dòng trong t dòng sau chứa một chuỗi s .

Kết quả: Đưa ra file văn bản SCORE.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng số nguyên.

Ví dụ:

SCORE.INP
2
JEROEN
JAN

SCORE.OUT
56
23



BẢNG XẾP HẠNG

Tên chương trình: RANKINGS.???

Cuộc thi lập trình đồng đội hàng năm có n đội tham gia, đánh số từ 1 đến n . Cuối mỗi đợt thi đều có bảng phân loại các đội theo kết quả đạt được. Bảng phân loại xếp hạng các đội năm trước đã được công bố. Năm nay Ban Giám khảo quyết định công bố kết quả phân loại một cách tế nhị hơn, không làm các đội đứng cuối quá buồn nản. Sau khi chấm điểm và sắp xếp các đội theo kết quả thi, Ban Giám khảo công bố danh sách tất cả các cặp đội có trình tự tương đối với nhau trong bảng mới khác với trình tự tương đối ở năm trước. Ví dụ, năm ngoái đội 13 đứng trên đội 6, nhưng năm nay đội 6 đứng trên đội 13, khi đó cặp số (6, 13) được thông báo. Thông tin này cho các đội biết họ đã tiến bộ như thế nào so với các đối thủ cũ, nhưng không xác định rõ thứ hạng của đội trong bảng tổng sắp.

Tất nhiên, biết bảng tổng sắp năm ngoái, các đội đều cố gắng xây dựng bảng tổng sắp năm nay. Không loại trừ khả năng có sai sót trong thông tin được công bố. Điều này cũng có thể làm rõ được.

Yêu cầu: Cho n ($2 \leq n \leq 500$), thứ tự t_i của đội i trong bảng tổng sắp năm ngoái ($1 \leq t_i \leq n$, $i = 1 \div n$), m – số cặp đội trong bảng thông báo ($0 \leq m \leq 25\,000$) và m cặp giá trị (a_j, b_j) $1 \leq a_j, b_j \leq n$, mỗi cặp xuất hiện một lần trong danh sách công bố. Hãy xác định bảng tổng sắp năm nay. Nếu ở vị trí nào đó không thể xác định đơn trị là vị trí của đội nào thì đưa ra dấu “?” ở vị trí tương ứng trong bảng. Nếu thông tin công bố có mâu thuẫn thì đưa ra thông báo “**IMPOSSIBLE**”.

Dữ liệu: Vào từ file văn bản RANKINGS.INP:

- Dòng đầu tiên chứa số nguyên k – số lượng tests ($1 \leq k \leq 100$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa số nguyên n ,
 - Dòng thứ 2 chứa n số nguyên t_1, t_2, \dots, t_n ,
 - Dòng thứ 3 chứa số nguyên m ,
 - Dòng thứ j trong m dòng tiếp theo chứa 2 số nguyên a_j và b_j .

Kết quả: Đưa ra file văn bản RANKINGS.OUT, kết quả mỗi test đưa ra trên một dòng – bảng tổng sắp xác định được hoặc thông báo “**IMPOSSIBLE**”.

Ví dụ:

RANKINGS.INP									
3									
5									
5	4	3	2	1					
2									
2	4								
3	4								
3									
2	3	1							
0									
4									
1	2	3	4						
3									
1	2								
3	4								
2	3								

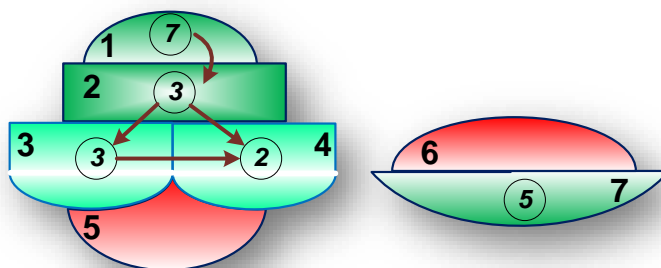
RANKINGS.OUT									
5	3	2	4	1					
2	3	1							
IMPOSSIBLE									



MẠO HIỂM

Tên chương trình: RISK.???

Mạo hiểm là trò chơi online. Bản đồ thế giới được chia thành các vùng bằng các đường biên giới. Mỗi vùng có thể do bạn hoặc đối phương của bạn kiểm soát. Mỗi vùng do bạn kiểm soát có một số nguyên dương đơn vị quân đội. Bạn có thể di chuyển quân đội để bố trí lại lực lượng. Một đơn vị quân đội có thể ở nguyên tại chỗ hoặc di chuyển sang vùng kề biên. Ở mỗi bước, từng đơn vị một di chuyển theo trình tự do bạn quyết định. Mỗi vùng mà bạn kiểm soát phải có ít nhất một đơn vị quân đội bảo vệ. Mục tiêu của bạn là phải tăng cường phòng thủ biên giới, tăng cường lực lượng ở các vùng kề với vùng của đối phương và đặc biệt là ở vùng biên hiện có lực lượng mỏng nhất (vùng yếu nhất).



Yêu cầu: Cho số vùng n ($1 \leq n \leq 100$), ma trận kề dưới dạng n xâu độ dài n chứa các ký tự thuộc tập $\{Y, N\}$, ký tự i trong xâu j bằng Y có nghĩa là các vùng i và j có biên chung, nếu ký tự này là N có nghĩa là 2 vùng không có biên chung. Cho các số a_1, a_2, \dots, a_n , trong đó a_i là số đơn vị quân đội trong vùng i ($0 \leq a_i \leq 100$). Nếu $a_i = 0$ – vùng i do đối phương kiểm soát. Hãy xác định số đơn vị tối đa có thể có sau một bước chuyển quân ở vùng yếu nhất.

Dữ liệu: Vào từ file văn bản RISK.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa số nguyên n ,
 - Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ,
 - Dòng thứ i trong n dòng tiếp theo chứa xâu độ dài n xác định ma trận kề.

Kết quả: Đưa ra file văn bản RISK.OUT, kết quả mỗi test đưa ra trên một dòng chứa một số nguyên – số đơn vị quân đội tối đa có thể có sau một bước chuyển quân ở vùng yếu nhất.

Ví dụ:

RISK.INP
2
3
1 1 0
NYN
YNY
NYN
7
7 3 3 2 0 0 5
NYNNNNN
YNYNNNN
NYNYNNN
NYNYNNN
NNYYNNN
NNNNNNY
NNNNNNY

RISK.OUT
1
4

BÁN ĐẤT

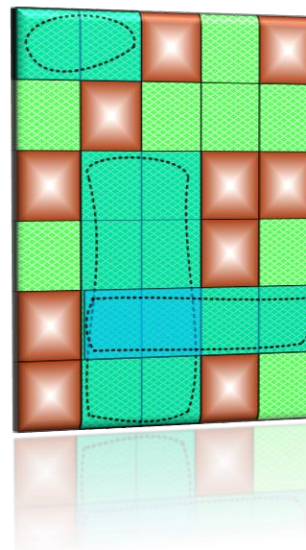
Tên chương trình: LAND.???

Giả như bạn biết ở xứ sở Những điều nghịch lý có bao nhiêu chuyện bất bình thường! Ví dụ toàn bộ diện tích được chia thành các ô vuông, mỗi ô là đồng cỏ hoặc đầm lầy, cai quản đất nước là một lớp nhân viên quan liêu và lười biếng. Nếu định mua một mảnh đất (gọi là một lô) bạn phải mua mảnh hình chữ nhật vì người ta không xem xét xử lý các lô dạng khác. Giá của lô đất được xác định theo chu vi của nó – các quan chức ngại phải thực hiện phép nhân.

Per có một mảnh đất hình chữ nhật kích thước $m \times n$ ô, trong đó một số ô là đồng cỏ, những ô khác – đầm lầy. Per muốn chia mảnh đất của mình thành các lô để bán. Dĩ nhiên, chẳng ai muốn mua ô đầm lầy. Khi bán, Per phải khai báo với các chức trách địa phương, các nhân viên hành chính sẽ báo giá theo chu vi của lô và ghi nhận tên người sở hữu vào ô dưới phải của lô.

Per nhận thấy có thể bán các lô có diện tích phủ lên nhau và thu lời nhiều hơn vì người ta chỉ quản lý sở hữu theo ô dưới phải. Với cách bán hợp lý Per đã thu được nhiều tiền nhất từ mảnh đất của mình, cụ thể là Per đã bán được p_i lô đất có chu vi i ($i = 4, 6, 8, \dots$).

Yêu cầu: Cho m, n ($1 \leq m, n \leq 1\,000$) và bản đồ mảnh đất dưới dạng m xâu độ dài n chỉ chứa các ký tự trong tập $\{., \#\}$, trong đó dấu chấm ứng với ô cỏ, ký tự $\#$ ứng với ô đầm lầy. Hãy đưa ra các $p_i \neq 0$, với mỗi p_i thông tin đưa ra có dạng $p_i \times i$ và đưa ra theo trình tự tăng dần của i .



Dữ liệu: Vào từ file văn bản LAND.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests ($1 \leq t \leq 100$),
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên trong nhóm chứa 2 số nguyên m và n ,
 - Dòng thứ i trong m dòng tiếp theo chứa xâu độ dài n xác định bản đồ mảnh đất.

Kết quả: Đưa ra file văn bản LAND.OUT, kết quả mỗi test đưa ra trên một nhóm dòng, mỗi dòng theo quy cách đã nêu.

Ví dụ:

LAND.INP
1
6 5
. . # . #
. # . . .
. . #
. . . # .
. . . .
. . # .

LAND.OUT
6 x 4
5 x 6
5 x 8
3 x 10
1 x 12



FILES ẢNH

Tên chương trình: JPGFILES.???

Steve có một máy chụp ảnh số và rất ham mê chụp ảnh. Steve lên danh sách thống kê các ảnh đã chụp, đánh số từ 1 đến n :

Số TT	Tên file
1	idx1001.jpg
2	idx1000.jpg
3	idx1014.jpg

Tên files có độ dài không quá 100 ký tự, bao gồm chữ cái la tinh (hoa hoặc thường), các ký tự “_” và “.”.

Có m dãy các bức ảnh liên tiếp nhau làm Steve đặc biệt thích thú vì nó gợi lại kỷ niệm đẹp của những ngày đã qua. Dãy thứ i bắt đầu từ ảnh ℓ_i đến ảnh r_i ($1 \leq \ell_i \leq r_i \leq n$). Steve ghi lại các số ℓ_i , r_i và in lại tên các files này để sau này dễ tìm lại chúng.

Yêu cầu: Cho n , m , tên các files, các số ℓ_i và r_i ($1 \leq n, m \leq 1\,000$). Hãy đưa ra danh sách tên các files được in.

Dữ liệu: Vào từ file văn bản JPGFILES.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ j trong n dòng sau chứa tên file thứ j ,
- Dòng tiếp theo chứa số nguyên m ,
- Dòng thứ i trong m dòng sau chứa 2 số nguyên ℓ_i và r_i .

Kết quả: Đưa ra file văn bản JPGFILES.OUT, mỗi dòng chứa một tên file.

Ví dụ:

JPGFILES.INP
3
idx1001.jpg
idx1000.jpg
idx1014.jpg
2
2 3
1 2

JPGFILES.OUT
idx1000.jpg
idx1014.jpg
idx1001.jpg
idx1000.jpg

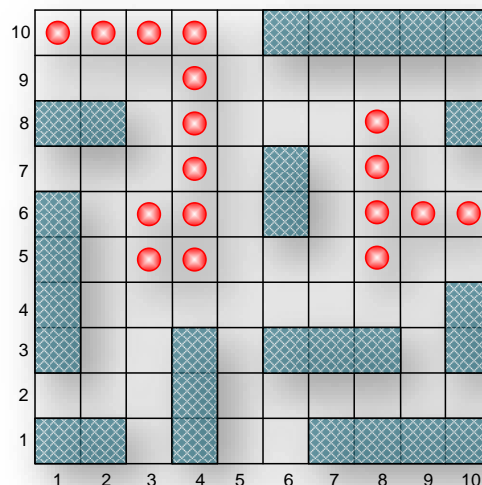


TÀU CHIẾN

Tên chương trình: SHIPS.???

Em người bạn của Steve rất thích trò chơi bắn tàu chiến. Nhưng trò chơi đó đã quá cũ và không tìm được ở đâu cả. Steve hứa sẽ viết lại chương trình trò chơi đó. Việc đầu tiên là phải lên sơ đồ bố trí tàu. Bản đồ bố trí tàu là lưới ô vuông kích thước $n \times m$ ($1 \leq n, m \leq 1000$). Tàu chiến là q ô tạo thành hình chữ nhật $1 \times q$ ô, nằm gọn trong bản đồ ($1 \leq q$).

Steve đã bố trí được k tàu, tàu thứ i được xác định bởi tọa độ các ô đầu và cuối $(x1_i, y1_i)$, $(x2_i, y2_i)$, trong đó hoặc $x1_i = x2_i$ hoặc $y1_i = y2_i$ ($1 \leq x1_i \leq x2_i \leq n$, $1 \leq y1_i \leq y2_i \leq m$, $0 \leq k \leq 10^4$, $i = 1 \div k$). Hai tàu chiến bất kỳ phải không có điểm chung, dù là trên cạnh hay ở đỉnh. Steve muốn kiểm tra lại xem các tàu chiến đã có thỏa mãn quy tắc bố trí hay không. Nếu hợp lý thì sẽ bố trí thêm một tàu kích thước 1×1 và tính số khả năng vị trí có thể đặt thêm tàu này.



Yêu cầu: Cho n, m, k và tọa độ $(x1_i, y1_i)$, $(x2_i, y2_i)$ của các con tàu. Hãy kiểm tra tính hợp lý của các tàu đã bố trí. Nếu không hợp lý thì đưa ra thông báo “**INCORRECT**”. Nếu hợp lý thì đưa ra số vị trí có thể bố trí thêm một tàu kích thước 1×1 .

Dữ liệu: Vào từ file văn bản SHIPS.INP:

- Dòng đầu tiên chứa 3 số nguyên n, m và k ,
- Dòng thứ i trong k dòng sau chứa 4 số nguyên $x1_i, y1_i, x2_i$ và $y2_i$.

Kết quả: Đưa ra file văn bản SHIPS.OUT thông báo “**INCORRECT**” hoặc một số nguyên – số khả năng bố trí tàu mới.

Ví dụ:

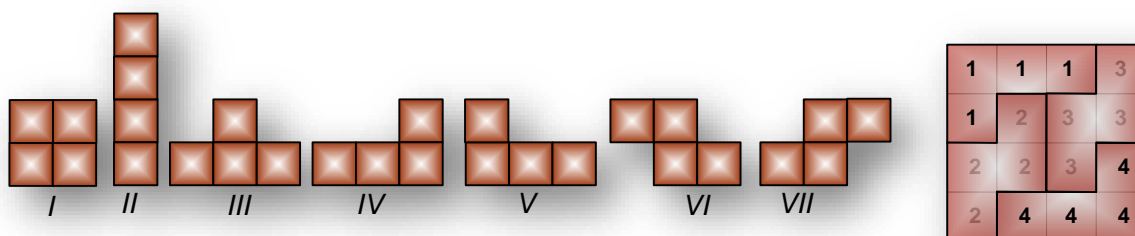
SHIPS.INP			
10	10	10	
1	1	1	2
1	4	3	4
1	7	1	10
3	1	6	1
3	6	3	8
3	10	4	10
6	6	7	6
8	1	8	2
8	10	8	10
10	6	10	10

SHIPS.OUT
17

TETRIS

Tên chương trình: TETRIS.???

Yan ham trò chơi điện tử Tetris đến mức bố mẹ phải cấm không cho sử dụng máy tính. Điều này cũng không làm Yan thất vọng. Cậu đã nghĩ ra trò chơi mới: “Tetris trên bàn”. Quy tắc trò chơi mới này rất đơn giản. Cho hình chữ nhật kích thước $n \times m$ ô. Ban đầu một số ô đã được tô màu, những ô còn lại – trống. Yan phải dùng các quân tetris lấp kín phần trống. Quân tetris làm bằng bìa, tô màu ở một mặt, vì vậy chỉ có thể xoay chứ không được lật, các quân tetris không được đặt chồng lên nhau và cũng không để lên các ô đã tô. Có 7 loại quân tetris. Ban đầu Yan định làm vô số quân mỗi loại nhưng Yan quá mệt mỏi sau khi làm được a_i quân loại i .



Yan vẽ một trường để chơi và muốn biết liệu với các quân tetris đã có liệu có thể lấp kín các ô trống hay không.

Yêu cầu: Cho n, m , các số a_i ($1 \leq n, m \leq 6, 0 \leq a_i \leq 10, i = 1 \div 7$) và bản đồ trường chơi. Hãy xác định xem có cách lấp kín các ô chưa tô hay không và đưa ra câu trả lời “**TAK**” hoặc “**NIE**” (Có hoặc Không, tiếng Balan), nếu có thì chỉ ra cách lấp. Các ô phủ bằng một quân tetris được đánh dấu bằng một số nguyên, các quân khác nhau – đánh số khác nhau, bắt đầu từ 1. Các ô ban đầu đã được tô đánh số là 0.

Dữ liệu: Vào từ file văn bản TETRIS.INP:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Dòng thứ 2 chứa 7 số nguyên a_1, a_2, \dots, a_7 ,
- Mỗi dòng trong n dòng sau chứa xâu độ dài m mô tả một dòng trong trường chơi, ký tự “.” Chỉ ô trống, ký tự “#” chỉ ô đã tô.

Kết quả: Đưa ra file văn bản TETRIS.OUT:

- Dòng thứ nhất chứa thông báo “**TAK**” hoặc “**NIE**”,
- Nếu có cách đặt quân thì bản đồ chỉ cách đặt gồm n dòng, mỗi dòng m số nguyên.

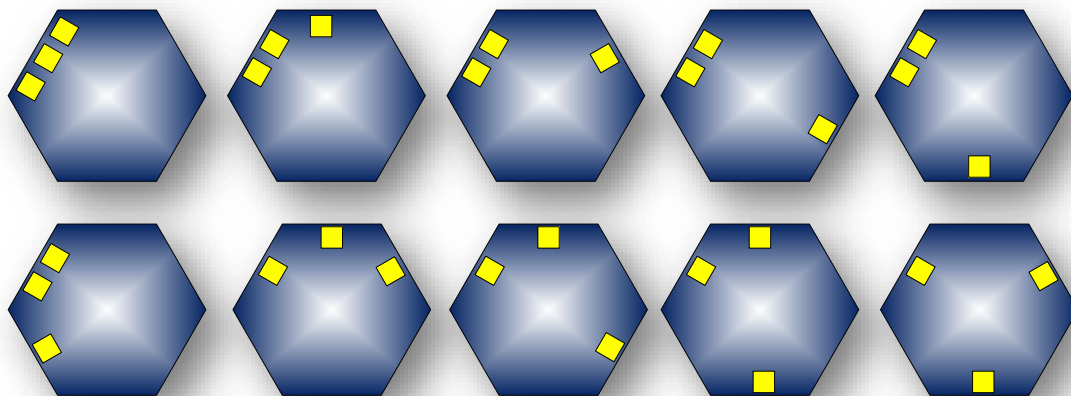
Ví dụ:

TETRIS.INP	TETRIS.OUT
4 4	TAK
1 1 0 3 0 2 0	1 1 1 3
.....	1 2 3 3
.....	2 2 3 4
.....	2 4 4 4
.....	



Những người da đỏ Maia ở thành phố Tikal đã xây dựng một đền thờ tuyệt đẹp, nền là hình đa giác đều n cạnh. Theo truyền thống, người ta mang vào trong đền k bức phù điêu giống nhau. Pháp sư chính của thành phố khẳng định phải đặt các bức phù điêu ở chân tường của đền. Mỗi buổi sáng Pháp sư lại vào đền thay đổi số phù điêu ở mỗi chân tường.

Khi đền thờ hoàn thành, thần núi hiện lên và cho biết khi cấu hình đặt phù điêu lặp lại, bầu trời sẽ sập xuống, nước biển dâng lên và đó là ngày tận thế. Thần núi còn cho biết rõ thêm, mọi cấu hình nhận được bằng cách xoay tháp sẽ bị coi là giống nhau.



Người dân thành phố cầu xin Pháp sư cho biết còn bao lâu nữa sẽ đến ngày tận thế. Vì số ngày có thể rất lớn nên Pháp sư thông báo cho nhân dân số ngày theo mô đun p , trong đó p là số nguyên tố.

Yêu cầu: Cho n , k và p ($3 \leq n \leq 5 \times 10^5$, $1 \leq k \leq 5 \times 10^5$, $10^6 < p < 10^9$). Đảm bảo p là nguyên tố. Hãy xác định số ngày theo mô đun p .

Dữ liệu: Vào từ file văn bản TIKAL.INP gồm một dòng chứa 3 số nguyên n , k và p .

Kết quả: Đưa ra file văn bản TIKAL.OUT một số nguyên – số ngày tính được (theo mô đun p).

Ví dụ:

TIKAL.INP
6 3 1000003

TIKAL.OUT
10

HẠ GIÁ

Tên chương trình: SALE.???

Để thu hút người tiêu dùng nhiều cửa hàng đã áp dụng các chính sách bán hạ giá khác nhau. Một mạng lưới bán văn phòng phẩm đã công bố “Mua n hàng giống nhau sẽ nhận thêm được một miễn phí!” và “Hãy mua k mặt hàng theo giá của $k-1$ mặt hàng!”

Ban Tổ chức Olympic cần in đề cho thí sinh. Sẽ phải tốn rất nhiều giấy cho việc này. Mỗi tập giấy giá b đồng. Ban Tổ chức có a đồng dành để mua giấy và muốn tìm cách mua được nhiều giấy nhất.

Yêu cầu: Cho n , k , a và b ($1 \leq n \leq 100$, $2 \leq k \leq 100$, $1 \leq a, b \leq 10^9$). Hãy xác định số tập giấy có thể mua được.

Dữ liệu: Vào từ file văn bản SALE.INP gồm một dòng chứa 4 số nguyên n , k , a và b .

Kết quả: Đưa ra file văn bản SALE.OUT một số nguyên – số tập giấy có thể mua được.

Ví dụ:

SALE.INP
4 4 13 2

SALE.OUT
8



GIẤY CHỨNG CHỈ

Tên chương trình: CERTIF.???

Steve tham gia rất nhiều kỳ thi, lớp ngoại khóa khác nhau và có đủ các loại chứng chỉ. Các chứng chỉ này được kẹp lưu trữ ở các tập khác nhau không theo một quy tắc nào cả. Cũng may là bên ngoài tập còn ghi số lượng chứng chỉ kẹp trong đó.

Hôm nay Steve cần đi ra văn phòng công chứng sao lại chứng chỉ kết quả thi Tin học Quốc gia để làm hồ sơ xin được tuyển thẳng vào khoa Công nghệ thông tin. Steve chỉ có một chứng chỉ này. Bạn ấy muốn tìm tập chứa chứa chứng chỉ đang cần, mang ra nơi công chứng và trong thời gian xếp hàng chờ đợi sẽ tìm và lấy nó ra để sao. Việc mở một tập kẹp chứng chỉ mất 1 giây, xem xét một chứng chỉ có phải là cái mình đang tìm hay không cũng mất 1 giây. Dĩ nhiên Steve không tìm ở các tập có ghi số lượng là 0. Việc chuyển từ tập này sang tập khác là không đáng kể.

Yêu cầu: Cho n – số tập lưu chứng chỉ và các số a_i – số chứng chỉ lưu trong tập i ($0 \leq a_i \leq 10^5$, $1 \leq n \leq 10^5$, $i = 1 \div n$). Hãy xác định, trong trường hợp xấu nhất, Steve cần ít nhất bao nhiêu thời gian để tìm ra tập cần thiết.

Dữ liệu: Vào từ file văn bản CERTIF.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản CERTIF.OUT một số nguyên – thời gian cần để tìm.

Ví dụ:

CERTIF.INP
4
1 0 2 1

CERTIF.OUT
4

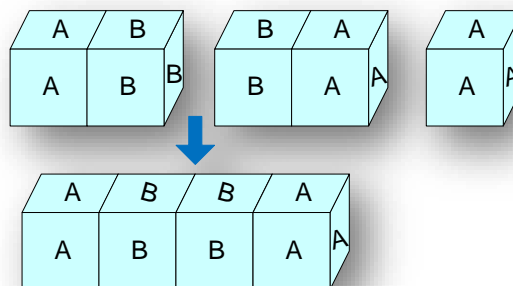


KHỐI LẬP PHƯƠNG

Tên chương trình: ABBAA.???

Jim là một chú bé, bình thường như bao chú bé khác. Jim còn rất bé nên chưa biết đếm. Một hôm, bố mẹ mang về cho chú bộ đồ chơi là các khối hộp kích thước $2 \times 1 \times 1$ có in chữ cái trên các mặt của khối lập phương đơn vị. Jim còn quá bé, bố mẹ không muốn làm chú rối trí nên chỉ chọn các khối hộp chỉ ghi 2 chữ cái đầu tiên là **A** và **B**.

Chú rất khoái chí và bắt đầu chơi, nhưng chơi không thật đúng cách: bẻ một số khối thành các hình lập phương để xếp nhà, sau đó, chú không thích các khối lập phương có chữ B và làm mất tất cả các khối này. Như vậy chú chỉ còn các khối hộp có ghi **AB** hoặc **BA** và một số khối lập phương có ghi chữ **A**. Lúc bố mẹ đi vắng cậu tìm được lọ keo dán. Là một cậu bé sáng ý, Jim nhanh chóng hiểu được tác dụng của vật này và dán các khối hộp cùng các khối lập phương thành các băng, băng thứ i có kích thước $1 \times 1 \times m_i$.



Khi bố mẹ về nhà Jim tự hào khoe kết quả lao động của mình. Qua mô tả rầm rối của Jim bố mẹ hiểu rằng Jim đã dán được n băng, băng thứ i biểu diễn xâu s_i (chỉ chứa các ký tự **A** và **B**). Bố mẹ Jim muốn biết mình đã hiểu đúng những gì Jim mô tả hay chưa, cụ thể là liệu có tạo được xâu s_i từ những gì mà Jim có hay không.

Yêu cầu: Cho n ($1 \leq n \leq 10$) và các xâu s_i ($i = 1 \div n$), tổng độ dài các xâu không vượt quá 10^5 . Với mỗi xâu hãy xác định là có thể tồn tại hay không và đưa ra câu trả lời “**YES**” hoặc “**NO**”.

Dữ liệu: Vào từ file văn bản ABBAA.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa xâu s_i .

Kết quả: Đưa ra file văn bản ABBAA.OUT các câu trả lời “**YES**” hoặc “**NO**”, mỗi câu trên một dòng.

Ví dụ:

ABBAA.INP	ABBAA.OUT
5	YES
A	NO
B	YES
ABBA	NO
BABBA	YES
ABBAA	

Harry và Rôn là người thường xuyên đọc blog của Steve. Theo yêu cầu của 2 bạn, Steve đã viết một chương trình chấm điểm. Harry rất thích nội dung của blog, Rôn thì ngược lại. Cách đây n ngày điểm của blog là x và hai bạn tranh luận với nhau xem sau n ngày điểm sẽ thay đổi như thế nào. Mỗi ngày nếu Harry bấm “+1” thì điểm đánh giá blog tăng thêm 1, nếu Rôn bấm “-1” thì điểm giảm đi 1. Chương trình chấm chưa hoàn thiện nên nếu một ngày nào đó cả 2 bạn cùng cho điểm thì điểm của blog sẽ là một số nguyên nào đó có giá trị tuyệt đối không vượt quá k .

Hôm nay Harry và Rôn quyết định chấm dứt cuộc tranh cãi về cách đánh giá blog, nhưng có một bạn thứ 3 xuất hiện, tham gia đánh giá làm chương trình cho điểm hoàn toàn rối loạn và ngừng hoạt động, điểm hiện nay của blog không hiển thị được. Harry và Rôn quyết định nếu một ngày nào đó đã qua điểm của blog lớn hơn m thì Harry thắng, nếu điểm blog nhỏ hơn $-m$ thì Rôn thắng. Dĩ nhiên, người nào đạt được mốc của mình trước thì người đó thắng.

Yêu cầu: Cho n, m, k, x, a_i và b_i , trong đó $a_i = 0$ nếu ngày thứ i Rôn không cho điểm và bằng 1 trong trường hợp ngược lại, $b_i = 0$ nếu ngày thứ i Harry không cho điểm và bằng 1 trong trường hợp ngược lại ($1 \leq n, m, k \leq 10^5, |x| \leq 1\,000, |x| \leq m, i = 1 \div n$). Hãy xác định số ngày tối thiểu cần thiết để Harry có thể thắng và số ngày tối thiểu cần thiết để Rôn có thể thắng. Nếu người nào không thể có khả năng thắng thì số ngày tối thiểu cần đưa ra sẽ là -1.

Dữ liệu: Vào từ file văn bản BLOG.INP:

- Dòng đầu tiên chứa các số nguyên n, m, k và x ,
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên a_i và b_i .

Kết quả: Đưa ra file văn bản BLOG.OUT hai số nguyên – số ngày tối thiểu cần thiết để Harry có thể thắng và số ngày tối thiểu cần thiết để Rôn có thể thắng.

Ví dụ:

BLOG.INP			
4	2	4	2
1	0		
1	0		
1	0		
1	0		

BLOG.OUT	
-1	-1

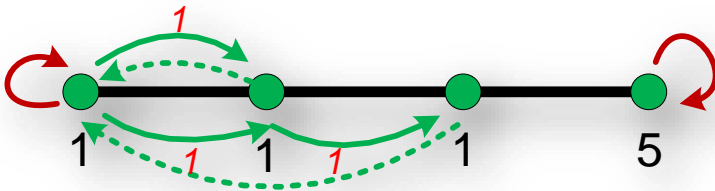


GIAO HÀNG

Tên chương trình: DELIVERY.???

Xứ Lineland có n thành phố, giữa thành phố thứ i và $i+1$ có đường nối hai chiều ($i = 1 \div n-1$). Mỗi lần vào thành phố i thì phải mua vé phí giao thông giá a_i . Tuy vậy cũng có thành phố không thu phí ($a_i = 0$), thậm chí có thành phố còn tặng thêm tiền để khuyến khích mọi người tới mua bán ($a_i < 0$).

Công ty giao hàng tại nhà có 2 trụ sở đóng ở thành phố thứ nhất và thành phố n . Có một đơn giao hàng tới n người ở n thành phố khác nhau. Mỗi chuyến đi xe chỉ chở được một mặt hàng cho một người. Việc giao hàng phải thực hiện theo nguyên tắc từ gần đến xa: chỉ được đi ngang qua các thành phố đã giao hàng. Xe của Công ty có một quyền ưu đãi: chỉ phải mua vé lượt đi, còn lúc về không phải mua vé (và cũng không nhận được tiền tặng). Xe ở trụ sở nào thì phải quay về trụ sở đó.



Yêu cầu: Cho n và a_i ($1 \leq n \leq 10^6$, $|a_i| \leq 10^4$, $i = 1 \div n$). Hãy xác định chi phí giao hàng nhỏ nhất.

Dữ liệu: Vào từ file văn bản DELIVERY.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản DELIVERY.OUT một số nguyên – chi phí giao hàng nhỏ nhất.

Ví dụ:

DELIVERY.INP
4
1 1 1 5

DELIVERY.OUT
3



d36 Int20110312 C

SIÊU XÂU CON

Tên chương trình: SUPERSUB.???

Xâu s được gọi là siêu chuỗi con của chuỗi t nếu tồn tại dãy chuỗi r_1, r_2, \dots, r_k thỏa mãn các điều kiện:

- $s = r_1 + r_2 + \dots + r_k$,
- r_i là chuỗi con các ký tự liên tiếp của t , $i = 1 \div k$.

Nếu s là siêu chuỗi con của t thì chuỗi r_1, r_2, \dots, r_k được gọi là lời giải. Lời giải tối ưu ứng với trường hợp k nhỏ nhất.

Yêu cầu: Cho chuỗi t và n chuỗi s_i , $i = 1 \div n$, độ dài của t và mỗi chuỗi s_i không vượt quá 10^6 , tổng độ dài các chuỗi không vượt quá 2×10^6 , chuỗi chỉ chứa các ký tự trong số 10 chữ cái thường đầu tiên của bảng chữ cái la tinh. Hãy xác định mỗi chuỗi s_i có phải là siêu chuỗi con của t hay không, nếu đúng – hãy chỉ ra một lời giải tối ưu. Lời giải được đưa ra dưới dạng các chuỗi con, nối với nhau bởi ký tự “|”. Nếu s_i không phải là siêu chuỗi con thì đưa ra thông báo “NO”.

Dữ liệu: Vào từ file văn bản SUPERSUB.INP:

- Dòng đầu tiên chứa chuỗi t ,
- Dòng thứ 2 chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa chuỗi s_i .

Kết quả: Đưa ra file văn bản SUPERSUB.OUT trên n dòng, dòng thứ i lưu kết quả xử lý chuỗi s_i .

Ví dụ:

SUPERSUB.INP
abacaba
3
cabab
dabacaba
aaaaaa

SUPERSUB.OUT
cab ab
NO
a a a a a a



CUỘC CHIẾN

Tên chương trình: BATTLE.???

Ở một thế giới xa xôi, nơi tận cùng của vũ trụ đang có một cuộc chiến giữa Ánh sáng và Bóng tối trong n lĩnh vực. Cuộc chiến diễn ra hàng ngày và cả 2 bên đều hoàn thiện dần khả năng chiến đấu trong từng lĩnh vực.

Ban đầu cuộc chiến, ở ngày số 0 trình độ chiến đấu trong lĩnh vực i của Ánh sáng là a_i , còn của bóng tối là s_i , sau mỗi ngày trình độ của Ánh sáng tăng thêm một lượng là b_i , còn Bóng tối – tăng thêm một lượng d_i .

Một lực lượng sẽ không bị lùi bước trước lực lượng khác nếu ở mỗi lĩnh vực đều có trình độ không thua kém đối phương.

Yêu cầu: Cho n, a_i, s_i, b_i, d_i ($1 \leq n \leq 100, 0 \leq a_i, s_i, b_i, d_i \leq 1\,000, i = 1 \div n$). Hãy xác định ngày đầu tiên Ánh sáng không bị lùi bước trước Bóng tối. Nếu ngày này không xảy ra thì đưa ra thông báo “**Strong is dark side of the force.**”

Dữ liệu: Vào từ file văn bản BATTLE.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ,
- Dòng thứ 3 chứa n số nguyên b_1, b_2, \dots, b_n ,
- Dòng thứ 4 chứa n số nguyên s_1, s_2, \dots, s_n ,
- Dòng thứ 5 chứa n số nguyên d_1, d_2, \dots, d_n .

Kết quả: Đưa ra file văn bản BATTLE.OUT một số nguyên xác định ngày đầu tiên Ánh sáng không bị lùi bước trước Bóng tối hoặc thông báo “**Strong is dark side of the force.**”

Ví dụ:

BATTLE.INP	
2	
0 0	
2 1	
2 3	
1 0	

BATTLE.OUT
3



DÂY NGOẶC

Tên chương trình: BRACKETS.???

Agnessa lần đầu tiên được làm quen với khái niệm biểu thức số học qua giờ Tin học. Cô bé quan tâm đến việc cái gì sẽ nhận được nếu ta bỏ hết các ký tự khác trong biểu thức ngoại trừ các ký tự ngoặc. Kết quả tìm kiếm trên mạng cho cô bé biết toán học gọi nó là dãy ngoặc và cô bé còn biết thêm thế nào là dãy ngoặc đúng.

Ví dụ $() (())$ là dãy ngoặc đúng vì nó có thể nhận được từ một biểu thức số học, chẳng hạn $(2+2):(3-(5-2)+4)$, còn các dãy ngoặc $((())$ hoặc $(())$ là không đúng. Dễ dàng thấy rằng, với 6 dấu ngoặc mở và đóng, trong đó có 3 ngoặc mở và 3 ngoặc đóng chỉ tồn tại 5 dãy ngoặc đúng: $((()))$, $((())())$, $(())(())$, $()(())()$ và $()()()$.

Agnessa thích thú tìm hiểu các phép biến đổi biểu thức ngoặc đúng, bắt đầu từ việc thêm ngoặc. Cô bé mau chóng nhận ra rằng nếu thêm một ngoặc thì dãy ngoặc không còn đúng, còn nếu thêm 2 ngoặc thì có thể có số dãy vẫn đúng. Ví dụ, với dãy ngoặc đúng $()()$, nếu thêm 2 dấu ngoặc ta có thể nhận được các dãy ngoặc đúng $((())())$, $((()))()$, $()(())()$, và $()()()$. Dễ dàng nhận thấy rằng chỉ có thể nhận được dãy ngoặc đúng nếu thêm một dấu ngoặc mở và một dấu ngoặc đóng. Ví dụ, từ dãy ngoặc đúng đơn giản nhất $()$ ta có 7 cách thêm 2 dấu ngoặc để có dãy ngoặc đúng mới $((()))$, $((())())$, $(())(())$, $()(())()$, $()()()$, $()()()$, $()()()$. Các dấu ngoặc thêm vào được in đậm.

Trong dãy ngoặc mới dấu mở ngoặc mới nằm ở vị trí i và dấu ngoặc đóng nằm ở vị trí j . Hai cách thêm ứng với các cặp (i_1, j_1) và (i_2, j_2) gọi là khác nhau nếu $i_1 \neq i_2$ hoặc $j_1 \neq j_2$.

Yêu cầu: Cho dãy ngoặc đúng độ dài $2n$ ($1 \leq n \leq 50\,000$). Hãy xác định số lượng cách thêm khác nhau 2 ngoặc để nhận được dãy ngoặc đúng độ dài $2n+2$.

Dữ liệu: Vào từ file văn bản BRACKETS.INP gồm một dòng chứa dãy ngoặc đúng độ dài $2n$.

Kết quả: Đưa ra file văn bản BRACKETS.OUT một số nguyên – số lượng cách thêm khác nhau 2 ngoặc để nhận được dãy ngoặc đúng độ dài $2n+2$.

Ví dụ:

BRACKETS.INP

BRACKETS.OUT

()

7



TRƯỜNG ĐÀO TẠO

Tên chương trình: SCHOOL.???

Để chuẩn bị cho World Cup 2018 người ta tổ chức một trường đào tạo để chuẩn bị nguồn cho đội tuyển Quốc Gia. Trường sẽ huấn luyện cho m thanh niên sinh các năm từ 1994 đến 1996. Có n người nộp đơn dự tuyển. Sau khi thực hiện các bài thi mỗi người có một điểm số đánh giá năng khiếu và trình độ chuyên môn. Điểm số của mỗi người khác nhau từng đôi một. Nhà trường dự định chọn a người sinh năm 1994, b người sinh năm 1995 và c người sinh năm 1996 ($a+b+c=m$). Ngoài ra còn phải thỏa mãn điều kiện: điểm nhỏ nhất của người sinh năm 1994 phải lớn hơn điểm nhỏ nhất của người sinh năm 1995, điểm nhỏ nhất của người sinh năm 1995 phải lớn hơn điểm nhỏ nhất của người sinh năm 1996, tất cả những người cùng năm sinh có điểm lớn hơn điểm nhỏ nhất của người cùng năm sinh được chọn sẽ phải được chọn.

Yêu cầu: Cho a, b, c, n và và điểm của mỗi người dự thi tuyển. Gọi số người được tuyển sinh năm 1994 là m_{94} , số người được tuyển sinh năm 1995 là m_{95} , số người được tuyển sinh năm 1996 là m_{96} ($m_{94}+m_{95}+m_{96} = m$). Hãy xác định số người được chọn sao cho $F = |m_{94}-a|+|m_{95}-b|+|m_{96}-c|$ là nhỏ nhất. Mỗi lứa tuổi phải có ít nhất một người được chọn.

Dữ liệu: Vào từ file văn bản SCHOOL.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên 3 số nguyên a, b, c ($a+b+c \leq n$),
 - Dòng thứ 2 chứa số nguyên n , tổng các n trong t tests không vượt quá 300 000,
 - Mỗi dòng trong n dòng sau chứa 2 số năm sinh và số điểm của mỗi người (xem ví dụ), điểm là số nguyên trong phạm vi từ 1 đến 10^9 .

Kết quả: Đưa ra file văn bản SCHOOL.OUT, nếu không có cách chọn thì đưa ra số -1, nếu có cách chọn thì đưa ra F và các giá trị m_{94}, m_{95}, m_{96} . Nếu tồn tại nhiều cách chọn khác nhau thì đưa ra một cách chọn bất kỳ.

Ví dụ:

SCHOOL.INP
1
2 3 1
7
1996 2
1994 7
1994 4
1996 1
1995 3
1994 5
1995 6

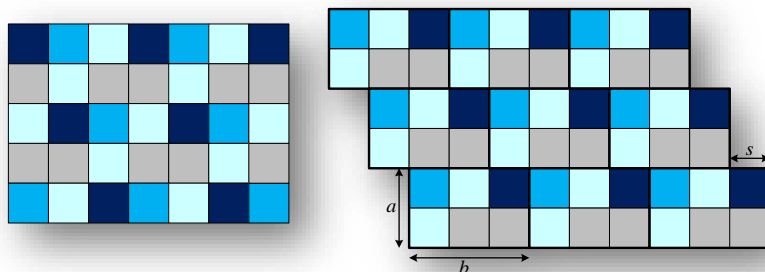
SCHOOL.OUT
2 3 2 1

TÀU CON THOI

Tên chương trình: SHUTLLE.???

Đề kỷ niệm mốc 50 năm con người bay vào vũ trụ người ta quyết định xây dựng một tàu con thoi thể hệ mới. Phần diện tích hình chữ nhật của vỏ tàu (ta sẽ gọi ngắn gọn là hình chữ nhật) phải được lát bằng các miếng chịu nhiệt hình vuông cùng kích thước. Có k loại miếng, mỗi loại đặc trưng bằng một ký tự la tinh in hoa trong số k chữ cái đầu tiên của bảng chữ cái la tinh.

Việc gắn trực tiếp từng miếng chịu nhiệt lên vỏ tàu là rất khó khăn, vì vậy người ta gắn chúng thành các tấm hình chữ nhật kích thước $a \times b$, sau đó gắn các tấm này vào hình chữ nhật theo từng hàng, từ trên xuống dưới. Các tấm phải được gắn theo cùng một chiều, nghĩa là nếu ta tịnh tiến song song một tấm cho trùng với tấm khác thì màu các miếng trên 2 tấm đó phải trùng nhau. Các tấm có thể nhô ra khỏi biên của hình chữ nhật. Khi gắn, hàng dưới có thể được tịnh tiến so với hàng trên một khoảng là s sang phải ($0 \leq s < b$).



Kỹ sư trưởng muốn chuẩn bị các tấm sao cho diện tích của tấm là nhỏ nhất.

Yêu cầu: Cho r, c – kích thước hình chữ nhật (theo đơn vị miếng chịu nhiệt), bản đồ các loại miếng chịu nhiệt cần gắn dưới dạng r xâu độ dài c chỉ chứa các ký tự in hoa đầu tiên của bảng chữ cái latin. Hãy xác định a, b và r tương ứng với tấm có diện tích nhỏ nhất. Dữ liệu đảm bảo tồn tại a và b thỏa mãn $2a \leq r$ và $2b \leq c$.

Dữ liệu: Vào từ file văn bản SHUTLLE.INP:

- Dòng đầu tiên chứa 2 số nguyên r và c ($2 \leq r, c \leq 1961$),
- Mỗi dòng trong r dòng sau chứa xâu độ dài c mô tả một dòng của hình chữ nhật.

Kết quả: Đưa ra file văn bản SHUTLLE.OUT trên một dòng 3 số nguyên a, b và s .

Ví dụ:

SHUTLLE.INP
5 7
DCADCAD
BABBABB
ADCADCA
BBABBAB
CADCADC

SHUTLLE.OUT
2 3 1



CÔNG VIÊN GIẢI TRÍ

Tên chương trình: ATTRACT.???

Ô tô buýt chở đoàn n học sinh tham dự Olympic Tin học tới công viên giải trí của thành phố vào thời điểm 0. Trong công viên có gian phòng trò chơi với m máy. Ở mỗi thời điểm mỗi máy chỉ dành cho một người chơi. Trò chơi ở máy thứ i kéo dài t_i đơn vị thời gian và người chơi không được phép ngắt giữa chừng.

Ban Tổ chức phải bố trí lịch chơi sao cho mỗi học sinh được chơi trên tất cả các máy và sau đó chở học sinh về được sớm nhất. Thời gian chuyển từ ô tô buýt vào phòng chơi và thời gian chuyển từ máy này sang máy khác là không đáng kể. Tại mỗi thời điểm một học sinh có thể chơi trên máy hoặc đi dạo công viên, chờ đợi đến lượt mình.

Yêu cầu: Cho n , m và t_i ($1 \leq m \leq n \leq 100$, $1 \leq t_i \leq 100$, $i = 1 \div m$). Hãy chỉ ra thời điểm ô tô buýt khởi hành ra về và với mỗi học sinh – chỉ ra thời điểm được bắt đầu chơi trên máy thứ i ($i = 1 \div m$).

Dữ liệu: Vào từ file văn bản ATTRACT.INP:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Dòng thứ 2 chứa m số nguyên t_1, t_2, \dots, t_m .

Kết quả: Đưa ra file văn bản ATTRACT.OUT: dòng đầu tiên chứa một số nguyên – thời điểm ô tô buýt khởi hành ra về, tiếp theo là n nhóm dòng, nhóm thứ j tương ứng với học sinh j , mỗi nhóm bắt đầu bằng một dòng trống, sau đó m dòng, mỗi dòng chứa 2 số nguyên k và v cho biết học sinh này bắt đầu chơi trên máy k từ thời điểm v , các dòng trong nhóm được sắp xếp theo thứ tự tăng dần của v .

Ví dụ:

ATTRACT.INP	
3	2
2	1

ATTRACT.OUT	
6	
1	0
2	2
1	2
2	4
2	0
1	4

