

Lập trình thi đấu 2022

Dăm ba Cấu Trúc Dữ Liệu

vjudge.net

2022

rủ người khác chơi game là tội ác

MAP-SET

417.

Chúng tôi phát triển một chương trình mã hóa đơn giản, mã hóa các từ có ít hơn năm chữ cái thành một số nguyên.

Xét trên bảng chữ cái tiếng Anh {a, b, c, ..., z}. Trên bảng chữ cái này, xác định tập các từ hợp lệ. Trong từ hợp lệ, các chữ cái được sắp xếp theo chiều từ trái qua phải theo thứ tự từ điển. Ví dụ:

abc aep gwz là ba từ hợp lệ, trong khi aab cat thì không.

Người ta gán số thứ tự cho mỗi từ hợp lệ, theo quy tắc sau:

```
a -> 1
b -> 2
.
.
z -> 26
ab -> 27
ac -> 28
.
.
az -> 51
bc -> 52
.
.
vwxyz -> 83.681
```

Chương trình của bạn đọc vào một loạt các từ có từ 1-5 chữ cái (mỗi từ trên một dòng). Đối với mỗi từ đọc vào, nếu từ đó là không hợp lệ, in ra số 0. Nếu từ hợp lệ, in ra số thứ tự của từ này.

INPUT

Đầu vào bao gồm một loạt các từ đơn, một trong mỗi dòng.

OUTPUT

Với mỗi từ, in ra đáp số cần tìm

Sample Input	Sample Output
z	26
a	1
cat	0
vwxyz	83681

484.

Cho một dãy các số nguyên. Xóa bỏ các ký tự trùng lặp, và in ra danh sách các số nguyên khác nhau trong dãy, và số lần xuất hiện của các số nguyên đó.

INPUT

Một danh sách các số nguyên (số dương, âm, và / hoặc 0).

OUTPUT

Nhiều dòng, mỗi dòng i là 1 cặp số nguyên (ai, bi), trong đó bi là số lần xuất hiện của ai trong dãy, các giá trị ai khác nhau và xuất hiện theo thứ tự xuất hiện trong dãy ban đầu.

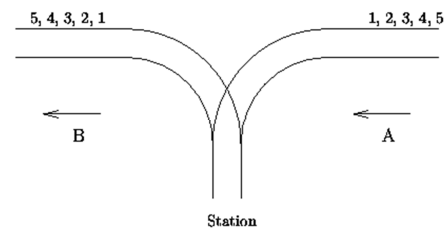
Sample Input	Sample Output
3 1 2 2 1 3 5 3 3 2	3 4 1 2 2 3 5 1

STACK

514. RAILS

Có một nhà ga nổi tiếng ở thành phố PopPush, là một vùng đồi núi. Trạm được xây dựng vào thế kỷ trước. Thật không may, quỹ đầu tư vào thời gian đó rất hạn chế. Bởi vậy chỉ có một đường ray duy nhất. Hơn nữa, nó lại dẫn vào nhà ga và kết thúc (xem hình) và do thiếu không gian, nên người ta chỉ xây có một đường ray.

Truyền thống của vùng này là tất cả các tàu đến từ hướng A và tiếp tục đi đến hướng B với thứ tự các toa tàu được sắp xếp lại theo một cách nào đó.



Giả sử rằng đoàn tàu đến từ hướng A có $N \leq 1000$ toa tàu với thứ tự các toa ban đầu được đánh số tăng dần 1, 2, ..., N. Trưởng đoàn tàu muốn biết đoàn tàu của họ có thể sắp xếp các toa tàu theo một thứ tự cho trước để tiếp tục đi theo hướng B được không?. Hãy giúp anh ta và viết một chương trình xác định xem có thể sắp xếp lại đoàn tàu theo thứ tự cần thiết không? Có thể định rằng các toa tàu có thể tự ngắt nối với các toa khác và tự di chuyển và nhà ga và các toa tàu có thể tự di chuyển đến hướng B. Cũng có thể giả định số lượng các toa tàu được đặt trong nhà ga là không giới hạn. Nhưng một khi một toa tàu đã di chuyển vào nhà ga nó không thể quay trở lại đường ray A và một khi nó đã rời nhà ga và đi theo B hướng nó không thể trở lại nhà ga được nữa.

INPUT

Input bao gồm nhiều khối dòng. Mỗi khối dòng ngoại trừ dòng cuối cùng mô tả một đoàn tàu và thứ tự cần sắp xếp lại. Trong dòng đầu tiên của khối là một số nguyên N mô tả như ở trên. Trong mỗi dòng tiếp theo của khối là một hoán vị của 1, 2, ..N. Dòng cuối cùng của khối là số 0. Khối cuối cùng bao gồm một dòng chứa một số 0 duy nhất.

OUTPUT

Các tập tin output chứa các dòng tương ứng với các dòng với hoán vị trong tập tin đầu vào. Một dòng của output ra chứa Yes nếu có thể sắp xếp các toa tàu theo thứ tự tương ứng với thứ tự trong input. Nếu không thì in No. Ngoài ra, có một dòng trống sau mỗi khối tương ứng với input. Với khối cuối cùng của input thì không in ra gì.

Sample Input	Sample Output
5	Yes
1 2 3 4 5	No
5 4 1 2 3	Yes
0	
6	
6 5 4 3 2 1	
0	
0	

673. PARENTHESES BALANCE

Bạn có được một dãy () và []. Dãy như thế được gọi là *đúng*, nếu:

- Nó là chuỗi rỗng
- Nếu dãy A và B đúng, thì dãy AB đúng,
- Nếu A đúng, thì (A) và [A] đúng.

Viết một chương trình kiểm tra xem 1 xâu có đúng hay không. Độ dài xâu nhỏ hơn 128.

INPUT

Dòng đầu tiên chứa số nguyên n là số test. n dòng tiếp theo là các test, mỗi dòng là 1 chuỗi () và [],

OUTPUT

Với mỗi test, ghi ra Yes hoặc No trong file output.

Sample Input	Sample Output
3	Yes
([])	No
(([])))	Yes
([][])()	

732. ANAGRAMS BY STACK

Làm cách nào sử dụng cấu trúc STACK để thực hiện phép đảo chữ? Có hai chuỗi thao tác trên STACK để đảo chữ TROT thành TORT

```
[
iiiioooo
ioiooio
]
```

trong đó i thể hiện thao tác Push and o thể hiện thao tác Pop. Cho 2 từ, bạn phải viết chương trình xuất ra 1 dãy các thao tác trên Stack để biến đổi từ thứ nhất sang từ thứ hai.

INPUT

Input có nhiều test. Mỗi test gồm hai dòng, mỗi dòng ghi 1 từ. Từ đầu là từ gốc, từ thứ 2 là từ đã bị đảo.

OUTPUT

Với mỗi test, in ra các dãy các chuỗi chữ i và o (dãy đã sắp xếp, theo thứ tự từ điển) để đảo chữ thứ nhất thành chữ thứ hai. Danh sách các dãy đặt giữa hai dấu [và]. Mỗi chuỗi chữ trong dãy nằm trên một dòng riêng. Trong một chuỗi chữ, các chữ i và o cách nhau bởi 1 ký tự dấu cách.

CHÚ Ý

Stack là một cấu trúc dữ liệu dùng để lưu trữ và có hai thao tác:

- Push – Nhét 1 đối tượng vào đỉnh ngăn xếp
- Pop – Lấy đối tượng ở đỉnh ngăn xếp ra.

Ngăn xếp ban đầu là trống.

Sample Input	Sample Output
madam	[
adamm	iiiiooooio
bahama	iiiiooooio
bahama	ioioioioio
long	ioioioioio
short]

eric	[
rice	i o i i i o o i i o o o
	i o i i i o o o i o i o
	i o i o i o i i i o o o
	i o i o i o i o i o i o
]
	[
]
	[
	i i o i o i o o
]

11111. GENERALIZED MATRIOSAKAS

Vladimir làm matrioshkas trong nhiều năm, matrioshkas những con búp bê lồng vào nhau – là sản phẩm thủ công đặc sắc của Nga. Matrioshka là một con búp bê có thể được tách thành hai nửa, bên trong chứa một con búp bê. Sau đó, con búp bê này có thể được tách ra để bên trong lại chứa một con búp bê khác. Điều này có thể được lặp đi lặp lại nhiều lần, cho đến khi không thể tách ra con búp bê cuối cùng bé nhất.

Gần đây, Vladimir nhận ra rằng có thể tổng quát ý tưởng của con búp bê lồng nhau thành món đồ chơi lồng nhau. Thật vậy, ông đã thiết kế những đồ chơi có chứa nhiều đồ chơi hơn. Một trong những món đồ chơi có thể được tách thành hai nửa và bên trong có thể có nhiều món đồ chơi bên trong. Đó là tính năng mới mà Vladimir muốn giới thiệu trong dòng sản phẩm đồ chơi của mình.



Vladimir đã phát triển một ký pháp để mô tả các đồ chơi lồng nhau. Một món đồ chơi được biểu diễn bằng một số nguyên dương, thể hiện kích thước của nó. Chính xác hơn: nếu khi mở món đồ chơi m , chúng ta thấy những đồ chơi n_1, n_2, \dots, n_r và $n_1 + n_2 + \dots + n_r < m$. Khi đó, chúng ta nói rằng đồ chơi m trực tiếp chứa đồ chơi n_1, n_2, \dots, n_r . Các đồ chơi nằm bên trong các đồ chơi n_1, n_2, \dots, n_r không được coi là trực tiếp nằm trong đồ chơi m .

Matrioshka tổng quát được thể hiện bằng một chuỗi các số nguyên dương có dạng:

$a_1 a_2 \dots a_N$

trong đó đồ chơi k được biểu diễn trong chuỗi với hai số nguyên $-k$ và k , với số âm xuất hiện trong chuỗi trước số dương – thể hiện 2 mảnh của một món đồ chơi k .

Ví dụ, chuỗi $-9 -7 -2 2 -3 -1 1 2 3 7 9$ biểu diễn một đồ chơi matrioshka tổng quát, cụ thể là có 6 món đồ chơi hai món đồ chơi 2; 1 3, 7 và 9 mỗi món có 1 đồ chơi. Lưu ý rằng: Đồ chơi 9 chứa trực tiếp đồ chơi 7, đồ chơi 7 chứa trực tiếp một món đồ chơi 2 và món 3; món đồ chơi ba chứa món đồ chơi 2, món đồ chơi 2 này chứa món đồ chơi 1. Chú ý rằng 2 mảnh -2 đầu tiên và 2 cuối cùng không phải hai mảnh của cùng một đồ chơi.

Mặt khác, các chuỗi sau đây không mô tả matrioshkas tổng quát:

$-9 -7 -2 2 -3 -1 -2 2 1 3 7 9$

bởi vì đồ chơi 2 lớn hơn so với đồ chơi 1 và không thể được nhét vào bên trong.

$-9 -7 -2 2 -3 -2 -1 1 2 3 7 -2 2 9$

bởi vì 7 và 2 không thể cùng nhét vào bên trong đồ chơi 9.

$-9 -7 -2 2 -3 -1 -2 3 2 1 7 9$

bởi vì có trục trặc trong việc nhét đồ vào đồ chơi 3.

Vấn đề của bạn là viết một chương trình để giúp đỡ Vladimir phân biệt một thiết kế tốt một thiết kế xấu.

INPUT

Các tập tin đầu vào có chứa nhiều test, mỗi test nằm trên một dòng riêng biệt. Mỗi test là một chuỗi các số nguyên khác không, có giá trị tuyệt đối nhỏ hơn 10^7 .

OUTPUT

Với mỗi test, in ra một dòng theo quy tắc sau :-> Matrioshka! nếu thiết kế mô tả một matrioshka tổng quát. Trong trường hợp khác, câu trả lời nên có dạng :-(Try again.

Sample Input	Sample Output
-9 -7 -2 2 -3 -2 -1 1 2 3 7 9	:-> Matrioshka!
-9 -7 -2 2 -3 -1 -2 2 1 3 7 9	:-> (Try again.
-9 -7 -2 2 -3 -1 -2 3 2 1 7 9	:-> (Try again.
-100 -50 -6 6 50 100	:-> Matrioshka!
-100 -50 -6 6 45 100	:-> (Try again.
-10 -5 -2 2 5 -4 -3 3 4 10	:-> Matrioshka!
-9 -5 -2 2 5 -4 -3 3 4 9	:-> (Try again.

QUEUE

540.

Queues và *Priority Queues* là các cấu trúc dữ liệu thông dụng. Nhưng *Team Queue*, - một cấu trúc dữ liệu hay gặp trong cuộc sống (như khi mọi người xếp hàng tại quầy lấy hàng ăn) lại không thông dụng bằng.

Ở team queue, mỗi phần tử thuộc một đội (team) nào đó. Nếu một phần tử vào hàng đợi, nó sẽ tìm kiếm trên hàng đợi từ đầu đến cuối để xem có đồng bọn (teammate) có trong queue chưa. Nếu có, nó sẽ đến xếp hàng sau đồng bọn đầu tiên. Nếu không, nó sẽ đến xếp hàng ngay sau. Ra khỏi hàng đợi giống như bình thường: các phần tử được xử lý từ đầu đến cuối hàng đợi.

Bạn phải viết chương trình mô phỏng hàng đợi kiểu này.

Input

Input có nhiều test. Mỗi test bắt đầu bằng số lượng team t ($0 \leq t \leq 1000$). Sau đó là t team, mỗi team bắt đầu bằng 1 số n là số lượng phần tử trong team, sau đó là n phần tử của team. Các phần tử là các số nguyên trong khoảng 0 - 999999. Một team có thể có tới 1000 phần tử.

Sau đó là danh sách các lệnh, Có 3 loại lệnh:

- ENQUEUE x - nhét phần tử x vào team queue
- DEQUEUE - lấy ra phần tử đầu tiên
- STOP - Kết thúc test

Input kết thúc khi $t=0$.

Output

Với mỗi test, in ra dòng "Scenario # k ", trong đó k là số thứ tự của test. Sau đó với mỗi lệnh DEQUEUE, in ra phần tử được lấy ra trên 1 dòng. In một dòng trống sau mỗi test, kể cả test cuối cùng.

Sample Input	Sample Output
2	101
3 101 102 103	102
3 201 202 203	103
ENQUEUE 101	201
ENQUEUE 201	202

ENQUEUE 102	203
ENQUEUE 202	
ENQUEUE 103	
ENQUEUE 203	
DEQUEUE	
DEQUEUE	
DEQUEUE	
DEQUEUE	
DEQUEUE	
DEQUEUE	
STOP	
0	
2	259001
5 259001 259002	259002
259003 259004	259003
259005	259004
6 260001 260002	259005
260003 260004	260001
260005 260006	
ENQUEUE 259001	
ENQUEUE 260001	
ENQUEUE 259002	
ENQUEUE 259003	
ENQUEUE 259004	
ENQUEUE 259005	
DEQUEUE	
DEQUEUE	
ENQUEUE 260002	
ENQUEUE 260003	
DEQUEUE	
DEQUEUE	
DEQUEUE	
DEQUEUE	
STOP	
0	

10172. The Lonesome Cargo Distributor

Các Plussians tin rằng Airport Based Cargo Distribution & Embarkation Facility (viết tắt là ABCDEF) tại các sân bay quốc tế Cosco (CIA- Cosco International Airport) là cơ sở lớn nhất trên toàn thế giới mặc dù lớn thứ hai ở Plussia. Máy bay chở hàng đi và đến từ nhiều quốc gia trên toàn thế giới (Cermany, QSA, Capan, Custralia ...). ABCDEF tham gia vào việc phân phối, bốc dỡ các kiện hàng những máy bay này chuyên chở.

Mỗi kiện hàng là một khối có kích thước cố định, trên đó có dán một thẻ dán nhãn tên nước cần chuyển hàng đến. Để thuận tiện, mỗi quốc gia có một định danh (Identification-ID) duy nhất. Ví dụ, nếu có n nước, mỗi nước có một ID duy nhất là một số nguyên khác nhau, từ 1 đến n .

Mỗi nước X có trạm hàng hoá riêng được xác định bởi ID quốc gia. Có hai cổng (cổng A và cổng B) ở mỗi trạm. Cổng A chứa những kiện hàng sẽ được chuyển bằng đường hàng không tới quốc gia X khi thuận tiện. Cổng B chứa những kiện hàng cần chuyển đến những quốc gia khác. Ga hàng hoá của các quốc gia nằm trên một hình tròn, nghĩa là nếu có n trạm thì các cặp trạm sau nằm cạnh nhau: $(1, 2)$, $(2, 3)$, $(3, 4)$, ..., $(n - 1, n)$ và $(n, 1)$.

ABCDEF có nhiều xe vận chuyển kiện hàng từ trạm này sang trạm khác. Tuy nhiên, các xe vận chuyển không thể đặt song song cạnh nhau. Một xe vận chuyển có thể chở nhiều hơn một kiện hàng hóa (mặc dù có giới hạn về số lượng kiện hàng tối đa mà xe vận chuyển có thể thực hiện) chỉ bằng cách đặt các kiện hàng chồng lên nhau. Xếp theo kiểu này có nghĩa bạn không thể lấy ra bất kỳ kiện hàng nào mình muốn. Ví dụ, để lấy ra kiện hàng thứ 3 từ trên, trước tiên bạn phải lấy ra hai kiện hàng trên cùng.

Một xe vận chuyển từ nhà ga này sang nhà ga kia theo vòng tròn, nghĩa là từ trạm 1, qua trạm 2, trạm 3,... trạm n rồi quay về trạm 1. Mất 2 phút để di chuyển giữa 2 trạm kề nhau.

Sau khi đến bất cứ trạm nào (chẳng hạn trạm X), xe vận chuyển cố gắng đỡ hàng xuống. Bắt đầu từ kiện hàng trên cùng, kiểm tra thẻ dán nhãn trên kiện hàng. Nếu có hàng hoá có nhãn X (là điểm đến), người ta sẽ bỏ kiện hàng này xuống cổng A, nếu không sẽ kiểm tra xem liệu hàng đợi ở cổng B có còn vị trí trống không, và nếu có thì sẽ đỡ kiện hàng này và đặt ở phía sau của hàng đợi. Thủ tục đỡ hàng này tiếp tục từ kiện hàng trên cùng xuống kiện hàng dưới cùng trên xe vận chuyển cho đến khi hoặc không thể làm được hoặc hết hàng trên xe vận chuyển (tùy theo điều kiện nào xuất hiện trước). Mỗi lần đỡ hàng thành công mất đúng 1 phút. Sau khi đỡ hàng xong, xe vận chuyển bắt đầu xếp hàng lên. Người ta lần lượt xếp lên xe vận chuyển các kiện hàng nằm ở hàng đợi trên cổng B cho đến khi hàng đợi không còn kiện hàng nào, hoặc khi xe vận chuyển đầy (tùy theo điều kiện nào xuất hiện trước). Mất đúng 4 phút để xếp một kiện hàng lên xe. Sau khi xếp xong, xe vận chuyển đi sang trạm kế tiếp trong vòng. Bằng cách này, trong nhiều năm, xe vận chuyển đã hoàn thành việc di chuyển hàng hoá từ cổng B sang cổng A của trạm thích hợp để chuyển về quốc gia thích hợp.

Tuy nhiên, sau một cuộc xung đột về việc chi trả lương, nhân viên của ABCDEF đã đình công vô thời hạn từ chủ nhật trước. Máy bay là đến và đi, nhưng không có ai để tải và dỡ hàng. Không có ai chuyển vận hàng hoá đến ga đích. Toàn bộ cơ sở tê liệt.

Tuy nhiên, bạn, luôn luôn bị đồng nghiệp ghét bỏ vì là cánh tay của ông chủ đã quyết định phá vỡ cuộc đình công và cứu lấy CIA. Bạn sẽ bắt đầu làm việc từ sáng ngày mai để phân phối, vận chuyển hàng hoá giữa các ga bằng xe vận chuyển. Ban đầu xe của bạn rỗng, bạn sẽ bắt đầu từ nhà ga 1 và tiếp tục di chuyển quanh vòng cho đến khi tất cả các hàng hoá đã được chuyển đến các trạm đích. Nhưng trước khi bắt đầu công việc, bạn phải quyết định viết một chương trình để xác định chính xác mất bao nhiêu thời gian để hoàn thành nó.

Input

Input có nhiều test. Dòng đầu tiên chứa số nguyên SET là số test. Theo sau là mô tả của SET test.

Dòng đầu tiên của test có 3 số nguyên : N, S và Q. N ($2 \leq N \leq 100$) là số trạm, S ($1 \leq S \leq 100$) là số kiện hàng tối đa mà xe trung chuyển có thể chở. Q ($1 \leq Q \leq 100$) là số lượng kiện hàng tối đa có thể đặt trong 1 hàng đợi ở cổng B (N hàng đợi này giống nhau).

Sau đó là N dòng, giả sử các dòng đánh số từ 1 đến N, với $1 \leq i \leq N$, dòng i chứa số nguyên Q_i ($0 \leq Q_i \leq Q$) là số kiện hàng tại hàng đợi của trạm thứ i, kế sau đó là Q_i số nguyên biểu diễn nhãn (đích ga đến) của các kiện hàng này. Có thể giả định không kiện hàng nào ở đây có đích đến là i.

Output

Với mỗi test, in ra thời gian để hoàn thành công việc.

Sample Input	Sample Output
2	72
5 2 3	72
3 4 5 2	11034.
2 1 3	
0	
3 3 5 1	
1 4	
5 2 3	
3 4 5 2	
2 1 3	
0	
3 3 5 1	
1 4	

11034

Trước khi cầu phở biến, bến phà được sử dụng để vận chuyển xe qua sông. Phà sông, không giống những phà to bây giờ, chạy men theo một dây dòng chằng giữa hai bờ sông. Xe ô tô lái xe lên phà từ một đầu, phà qua sông, và xe ô tô rời phà từ phía đầu bên kia.

Có một phà sông dài 1 mét. Một chiếc xe có thể đến một trong hai bờ sông và được phà vận chuyển sang bờ bên kia. Phà di chuyển liên tục giữa hai bờ, miễn là nó còn đang chở một chiếc xe hoặc có ít nhất một chiếc xe chờ đợi tại một bờ nào đó. Bất cứ khi nào phà ghé bờ, nó thả các xe trên đó xuống và các ô tô đang sắp hàng được

lên phà - miễn là trên phà còn chỗ. Những chiếc xe được lên phà theo thứ tự đến, các ô tô xếp hàng nối đuôi nhau trên phà trên cùng một hàng. Ban đầu phà ở bên bờ trái để sửa chữa. Trong khi đó, các xe ô tô kéo đến cả hai bờ để xếp hàng qua sông.

INPUT

Dòng đầu tiên chứa c, số test. Mỗi test bắt đầu với l, m. m dòng theo mô tả những chiếc xe theo thứ tự đến. Mỗi dòng ghi chiều dài của một chiếc xe hơi (tính theo cm), và bờ (trái hoặc phải) mà xe ô tô muốn sang ("left" hoặc "phải").

OUTPUT

Đối với mỗi test, in ra một dòng ghi số lần phà qua sông để phục vụ tất cả các xe đang xếp hàng đợi.

Sample Input	Sample Output
4	3
20 4	3
380 left	5
720 left	6
1340 right	
1040 left	
15 4	
380 left	
720 left	
1340 right	
1040 left	
15 4	
380 left	
720 left	
1340 left	
1040 left	
15 4	
380 right	
720 right	
1340 right	
1040 right	

PRIORITY QUEUE

1203. ARGUS

Có thể coi Luồng dữ liệu là chuỗi các cụm dữ liệu, ví dụ dữ liệu lấy từ các bộ cảm biến, lưu lượng trên Internet, các cuộc đấu giá trực tuyến... Hơn thế nữa, liên tục xuất hiện các truy vấn (câu hỏi) trên luồng dữ liệu và có dữ liệu mới bổ sung vào luồng dữ liệu. Ví dụ, một hệ thống giám sát nhiệt độ có thể nhận được những truy vấn như sau:

Query-1: Cứ mỗi 5 phút, trả lại nhiệt độ cao nhất trong 5 phút vừa qua

Query-2: Cứ mỗi 10 phút, trả lại nhiệt độ trung bình của mỗi tầng.

Argus là hệ thống quản lý luồng dữ liệu và xử lý các truy vấn trên luồng. Argus sẽ thực thi để đưa ra trả lời cho các truy vấn trên luồng dữ liệu luôn thay đổi thường xuyên.

Với Argus, chúng ta dùng lệnh sau để gửi một truy vấn:

Register Q_num Period. Q_num (0 < Q_num ≤ 3000) là mã số của truy vấn, và Period (0 < Period ≤ 3000) là khoảng thời gian giữa 2 lần trả lại kết quả liên tiếp nhau. Sau khi đăng ký Period giây, kết quả đầu tiên sẽ được trả về, và sau đó, cứ Period giây lại có một kết quả trả về. Chúng ta có một vài truy vấn khác nhau. Biết rằng

mọi truy vấn có giá trị Q_num khác nhau. Công việc của bạn là xác định K truy vấn đầu tiên để trả lời. Nếu tại cùng thời điểm có nhiều truy vấn phải trả lời, chúng sẽ được trả lời theo thứ tự tăng dần của Q_num.

INPUT

Phần đầu tiên là các truy vấn được gửi đến Argus,mỗi truy vấn nằm trên một dòng. Số lượng truy vấn không vượt quá 1000. Và giả sử mọi truy vấn được xử lý cùng lúc. Ký tự # đánh dấu kết thúc phần truy vấn. Phần thứ 2 là một số nguyên dương K(≤ 10000).

OUTPUT

XUất ra giá trị Q_num của K truy vấn phải trả lời đầu tiên.

Sample Input	Sample Output
Register 2004 200	2004
Register 2005 300	2005
#	2004
5	2004
	2005

10954. CỘNG TẮT.

Đây chỉ là bài toán cộng một tập hợp các số, nhưng khi thực hiện, xuất hiện chi phí của phép cộng. Để cộng 1 với 10, bạn cần một chi phí 11. Nhưng có vài cách để cộng 1,2 và 3 với những chi phí khác nhau.

1 + 2 = 3, chi phí 3	1 + 3 = 4, chi phí 4	2 + 3 = 5, chi phí 5
3 + 3 = 6, chi phí 6	2 + 4 = 6, chi phí 6	1 + 5 = 6, chi phí 6
Tổng chi phí: 9	Tổng chi phí: 10	Tổng chi phí: 11

Hãy cộng một tập hợp các số nguyên với chi phí bé nhất.

INPUT

Có nhiều test, bắt đầu mỗi test là số nguyên N (2 ≤ N ≤ 5000), sau đó là N số nguyên dương (< 100000). Input kết thúc khi N =0 và trường hợp này không phải xử lý.

OUTPUT

Với mỗi test in ra chi phí cộng tối thiểu.

Sample Input	Sample Output
3	9
1 2 3	19
4	
1 2 3 4	
0	

11995 . CẤU TRÚC DỮ LIỆU GÌ

Có một cấu trúc dữ liệu Túi với hai thao tác sau:

- 1 x : Ném phần tử x vào túi.
- 2 : Lấy một phần tử từ túi.

Cho một chuỗi các thao tác cùng giá trị trả về, bạn phải đoán cái Túi là Ngăn Xếp (Last-In, First-Out), Hàng Đợi (First-In, First-Out), Hàng Đợi Ưu Tiên (luôn lấy ra phần tử to nhất trước) hay Không Xác Định!

INPUT

Có nhiều test. Trong mỗi test, dòng đầu là số nguyên n ($1 \leq n \leq 1000$). Mỗi một trong n dòng kế tiếp hoặc ghi lệnh 1, hoặc ghi lệnh 2 (nếu là lệnh 2 thì sau lệnh 2 là một số nguyên x – là giá trị mà chúng ta lấy được từ Túi). Giá trị x luôn nhỏ hơn 100. Kích thước file input nhỏ hơn 1MB.

OUTPUT

Với mỗi test, in ra trên một dòng: stack, queue, priority queue, impossible (không phải là stack, queue, priority queue), not sure (có thể là hai hoặc nhiều cấu trúc dữ liệu) .

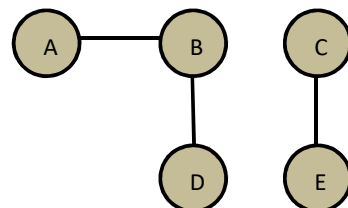
Sample Input	Sample Output
6	queue
1 1	not sure
1 2	impossible
1 3	stack
2 1	priority queue
2 2	
2 3	
6	
1 1	
1 2	
1 3	
2 3	
2 2	
2 1	
2	
1 1	
2 2	
4	
1 2	
1 1	
2 1	
2 2	
7	
1 2	
1 5	
1 1	
1 3	
2 5	
1 4	
2 4	

DISJOIN SET - UNION & FIND

459. ĐỒ THỊ LIÊN THÔNG

Đồ thị G được gọi là liên thông nếu có giữa hai đỉnh bất kỳ có một tuyến đường. Đồ thị dưới đây không liên thông vì không tồn tại đường đi giữa A và C .

Tuy nhiên Đồ thị này có chứa nhiều đồ thị con là thành phần liên thông, chẳng hạn các tập con sau đây $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, $\{A,B\}$, $\{B,D\}$, $\{C,E\}$, $\{A,B,D\}$



Thành phần liên thông được gọi là cực đại nếu như không thể thêm bất kỳ đỉnh và cạnh nào của đồ thị ban đầu vào mà không mất đi tính liên thông. Trong ví dụ trên $\{A, B, D\}$ và $\{C, E\}$ là hai thành phần liên thông cực đại.

Viết chương trình xác định số lượng thành phần liên thông cực đại của một đồ thị cho trước

Input and Output

Dòng đầu tiên chứa một số nguyên là số test. Sau đó là một dòng trống và các bộ test. Mỗi bộ test cách nhau bằng một dòng trống. Dòng đầu tiên trong mỗi bộ test là một chữ cái in hoa biểu diễn tên gọi lớn nhất (theo thứ tự từ điển) của các đỉnh đồ thị. Mỗi dòng kế tiếp là một cặp hai chữ cái in hoa biểu diễn một cạnh đồ thị. Kết thúc input là một dòng trống.

Với mỗi test, in ra số lượng thành phần liên thông cực đại. Mỗi test nằm trên một dòng.

Ví dụ

Sample Input	Sample Output
1	2
E	
AB	
CE	
DB	
EC	

793. KẾT NỐI MẠNG

Khuê phải quản trị một mạng máy tính. Khuê có trong tay nhật ký mạng, ghi lại kết nối giữa các cặp máy tính trong mạng. Mỗi kết nối là hai chiều. Hai máy tính có kết nối nếu hoặc chúng có kết nối trực tiếp với nhau, hoặc nếu chúng có kết nối tới cùng một máy tính thứ ba. Đôi khi Khuê sẽ phải dựa vào nhật ký để trả lời ngay lập tức – hai máy tính nào đó có kết nối (trực tiếp hay gián tiếp) với nhau không. Dựa trên file text nhật ký, hãy viết chương trình để đếm số lượng câu trả lời đúng hoặc sai cho dạng câu hỏi sau: liệu máy tính i có kết nối với máy tính j hay không?

Input and Output

Dòng đầu tiên chứa số lượng test. Sau đó là một dòng trống. Mỗi test bao gồm::

1. Số N là số lượng các máy tính có trong mạng;

2. Sau đó là các dòng, mỗi dòng là một cặp có dạng sau:

(a) c $computer_i$ $computer_j$, trong đó $computer_i$ và $computer_j$ là các số nguyên từ 1 đến N . thể hiện $computer_i$ và $computer_j$ **có kết nối**.

(b) q $computer_i$ $computer_j$, trong đó $computer_i$ và $computer_j$ là các số nguyên từ 1 đến N . đây là câu hỏi mà bạn phải trả lời **$computer_i$ có kết nối với $computer_j$ không?**

Giữa các test có một dòng trống. Nhật ký được cập nhật liên tục.

Hãy xuất ra $N1$ và $N2$, trong đó $N1$ là số câu trả lời CÓ và $N2$ là số câu trả lời KHÔNG.

Ví dụ

Sample Input	Sample Output
1	1,2
10	
c 1 5	
c 2 7	
q 7 1	
c 3 9	
q 9 6	

10178. ĐỒ THỊ PHẪNG

Đồ thị phẳng là đồ thị có thể vẽ trên mặt phẳng mà không có hai cạnh nào giao nhau (không tính việc giao nhau tại đỉnh).

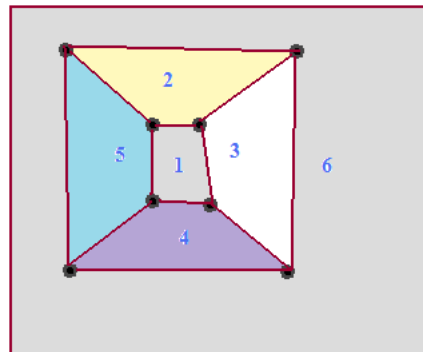
Đồ thị trên là một đồ thị phẳng. Đồ thị có 6 khu vực tô với 6 màu khác nhau, được đánh số lần lượt từ 1 đến 6. Bạn có một đồ thị phẳng và phải đếm số khu vực của đồ thị đó

Input

Input có nhiều test, mỗi test bắt đầu bằng một dòng ghi hai số nguyên N và E , tương ứng là số đỉnh và số cạnh của đồ thị. E dòng tiếp theo mô tả các cạnh của đồ thị, có khuôn dạng $n1\ n2$ trong đó $n1\ n2$ là các chữ cái Latinh thường, thể hiện giữa đỉnh $n1$ và đỉnh $n2$ có một cung.

Output

In ra số lượng khu vực của đồ thị



Sample Input	Sample Output
1 0	1
3 3	2
A B	
B C	
A C	

10227. RỪNG

Triết gia Berkeley đặt ra một câu hỏi thú vị : Nếu không có ai nghe, một cây đổ trong rừng có tạo ra âm thanh không ? Trong rừng có T cây đánh số từ 1 đến T và P người đánh số từ 1 đến P .

Input

Số đầu tiên là số lượng test, sau đó là một dòng trống. Mỗi test cách nhau một dòng trống. Mỗi test bắt đầu bằng 2 số nguyên P và T ($0 < P, T < 100$), sau đó là các dòng dạng $i\ j$, trong đó người i nghe thấy cây j đổ. Mọi người có thể có những ý kiến khác biệt về cây nào đổ.

Output

Xuất ra số lượng ý kiến khác biệt ? Hai người có cùng ý kiến khi và chỉ khi tập cây đổ mà họ nghe là giống hệt nhau

Sample Input	Sample Output
1	2
3 4	
1 2	
3 3	
1 3	
2 2	
3 2	
2 4	

10507. TỈNH THỨC

Các nghiên cứu mới nhất trong ngành sinh lý học thần kinh chỉ ra rằng bên trong não bộ, có những tuyến truyền dẫn rất lớn kết nối những khu vực khác nhau của não bộ. Hơn thế nữa, các nhà khoa học còn khám phá: nếu trong vòng một năm, một vùng não trong trạng thái ngủ có kết nối trực tiếp với ít nhất 3 vùng não trong trạng thái thức thì vùng não ngủ sẽ chuyển sang trạng thái thức (và sẽ ở lại trạng thái này). Giả sử các khu vực não bộ có tên gọi A, B, C,... và ban đầu, các khu vực đều ở trạng thái ngủ, một số khu vực có kết nối với nhau. Nếu kích thích trực tiếp để làm thức ba khu vực não nào đó, phải mất bao nhiêu năm thì toàn bộ các khu vực não chuyển sang trạng thái thức ?

Input

Có nhiều test, các test cách nhau bằng một dòng trống. Khuôn dạng các test như sau: Dòng đầu tiên là số nguyên $3 \leq N \leq 26$, là số lượng các khu vực trong trạng thái ngủ ban đầu. Dòng thứ hai là số nguyên dương M – số lượng kết nối giữa các khu vực (nếu A có kết nối với B thì B có kết nối với A). Dòng thứ 3 là ba ký tự thể hiện ba khu vực của bộ não bị kích thích trực tiếp. M dòng kế tiếp, mỗi dòng chứa 2 ký tự, thể hiện những đường kết nối trực tiếp giữa hai khu vực não.

Output

Xuất ra một trong hai dòng:

- “THIS BRAIN NEVER WAKES UP”
- “WAKE UP IN, “ n “, YEARS”, trong đó n là số năm sau đó tất cả các khu vực bộ não đều chuyển sang trạng thái thức.

Sample Input	Sample Output
6 11 HAB AB AC AH BD BC BF CD CF CH DF FH	WAKE UP IN, 3, YEARS

10583. ĐẾM TÔN GIÁO

Bạn có nhiệm vụ đếm trong trường học của mình, có bao nhiêu tôn giáo có tín đồ. Trường của bạn có n sinh viên ($0 < n \leq 50000$). Bạn không thể đến từng người để hỏi xem họ theo tôn giáo nào. Hơn thế nữa, nhiều sinh viên coi tôn giáo là vấn đề riêng tư và không muốn trả lời trực tiếp. Một cách giải quyết là hỏi m ($0 \leq m \leq n(n-1)/2$) cặp sinh viên xem họ có cùng chung một tôn giáo hay không. Từ lượng dữ liệu thu thập được, tuy chưa xác định được ai theo tôn giáo nào, nhưng bạn ước đoán được số lượng tôn giáo cực đại có tín đồ trong trường. Mỗi sinh viên chỉ theo nhiều nhất một tôn giáo.

Input

Có nhiều test. Dòng đầu tiên của mỗi test là hai số nguyên n và m. m dòng tiếp theo có dạng x y thể hiện sinh viên x và sinh viên y có cùng tôn giáo. Các sinh viên đánh số từ 1 đến n. Kết thúc khi n=m=0.

Output

Với mỗi test, in ra số lượng tôn giáo cực đại

Sample Input	Sample Output
10 9	Case 1: 1
1 2	Case 2: 7
1 3	
1 4	
1 5	
1 6	
1 7	
1 8	
1 9	
1 10	
10 4	
2 3	
4 5	
4 8	
5 8	
0 0	

10608. ĐẾM BẠN

Trong trường chuyên có N học sinh. Thừa ban đầu có một số cặp bạn trong trường. Tuy nhiên sau một hồi chơi chung, “bạn của bạn cũng là bạn của ta” - nghĩa là nếu A và B là bạn, B và C là bạn thì A và C cũng là bạn. Khuê có nhiệm vụ đếm nhóm bạn có số lượng lớn nhất.

Input

Dòng đầu là số nguyên thể hiện số test. Sau đó là các bộ test. Ở mỗi bộ test, dòng đầu ghi 2 số nguyên N ($1 \leq N \leq 30000$) và M ($0 \leq M \leq 500000$), tương ứng là số học sinh của trường và số cặp bạn bè ban đầu. M dòng tiếp theo ghi hai số A và B ($1 \leq A \leq N$, $1 \leq B \leq N$, $A \neq B$) mô tả A và B là bạn. (các dòng mô tả này có thể trùng lặp).

Output

Với mỗi test, in ra số lượng bạn bè trong nhóm bạn cực đại.

Sample Input	Sample Output
2	3
3 2	6
1 2	
2 3	
10 12	
1 2	
3 1	
3 4	
5 4	
3 5	
4 6	
5 2	
2 1	
7 10	
1 2	
9 10	
8 9	

10685. SINH TỒN THIÊN NHIÊN

Trong tự nhiên luôn luôn tồn tại các chuỗi thức ăn. Thực vật ở mức thấp nhất của chuỗi, các động vật nhỏ ăn thực vật và động vật to ăn động vật nhỏ. Có thể có chu trình trong chuỗi thức ăn, chẳng hạn như khi chết đi, xác động vật bị phân hủy thành mùn và trở thành chất bón (thức ăn) cho thực vật. Trong bài toán này, bạn có nhiệm vụ xác định chuỗi thức ăn lớn nhất giữa một nhóm sinh vật. Nếu A ăn B thì A và B thuộc về cùng một chuỗi.

Có nhiều test, mỗi test có khuôn dạng sau. Dòng đầu của test chứa hai số nguyên C ($1 \leq C \leq 5000$), và R ($0 \leq R \leq 5000$), tương ứng là số lượng sinh vật và số lượng các quan hệ trong nhóm. Sau đó là C dòng mô tả tên các sinh vật, tên sinh vật không quá 30 ký tự, gồm chữ cái Latin (chữ thường và chữ hoa) cùng dấu gạch dưới ($_$). Sau đó là R dòng mô tả các mối quan hệ gồm tên sinh vật 1 và tên sinh vật 2, ở đây sinh vật 2 ăn thịt sinh vật 1. Không có sinh vật nào ăn thịt chính mình.

Output

Với mỗi bộ test, in ra chuỗi thức ăn kích thước cực đại.

Sample Input	Sample Output
5 2 caterpillar bird horse elefant herb herb caterpillar caterpillar bird 0 0	3

11503. BẠN ẢO

Những ngày nay, bạn có thể làm mọi thứ online. Ví dụ bạn dùng Facebook để kết bạn ảo. Với một số người, phát triển mạng xã hội (bạn của mình, bạn của bạn, bạn của bạn của bạn,...) là một sở thích gây nghiện. Ai đó sưu tập tem – còn những người nghiện này sưu tập bạn ảo. Nhiệm vụ của bạn là quan sát dữ liệu lấy từ Facebook và xác định kích thước mạng xã hội của một ai đó. Quan hệ bạn bè là 2 chiều, tức là nếu Vô Phúc là bạn của Bất Hiểu thì Bất Hiểu cũng là bạn của Vô Phúc.

Input

Dòng đầu tiên là số test. Mỗi test bắt đầu bằng một số nguyên $F < 100\,000$ – là số lượng quan hệ bạn bè. F dòng tiếp theo, mỗi dòng chứa tên hai người (cách nhau bằng dấu cách) có quan hệ bạn bè. Tên người là xâu không quá 20 ký tự.

Output

Mỗi khi có một quan hệ tình bạn mới được thiết lập, in ra kích thước của mạng xã hội của hai người vừa mới kết bạn với nhau.

Sample Input	Sample Output
1	2
3	3
Fred Barney	4
Barney Betty	
Betty Wilma	

11690. TIỀN BẠC

Trường chuyên tổ chức cho học sinh khối 10 đi chơi ở vùng núi Ba Vì tuyệt đẹp. Khi đi chơi, mọi người có vay nợ nhau một số tiền để mua sắm, đánh tiền lên, tá lả, hát karaoke. Trong hai ngày ở đó, có quá nhiều sự kiện kinh khiếp đã diễn ra (không thể viết ra ở đây). Chỉ biết hậu quả là rất nhiều câu nói “Tớ không bao giờ chơi với cậu nữa !” đã được cất lên (tất nhiên câu này lịch sự).

Quay về Mễ Trì, nhóm học sinh lớp 10 phát hiện mình chưa thanh toán sòng phẳng các khoản nợ với nhau. Nhưng giờ đây, việc giải quyết nợ nần khó khăn hơn rất nhiều bởi vì nhiều học sinh còn không muốn chạm mặt nhau – nói chỉ đến việc trả nhau tiền.

Bạn được yêu cầu giúp đỡ, bạn hỏi được mọi người tổng số tiền người đó nợ (hay bị nợ) và người này còn làm bạn với ai. Từ tất cả những thông tin này, bạn có thể kiểm tra xem liệu nhóm học sinh khối 10 có thể thanh toán sòng phẳng cho nhau được không (tiền chỉ có thể trao đổi giữa những cặp học sinh vẫn còn là bạn).

Input

Dòng đầu tiên là số nguyên N – là số test, mỗi test có khuôn dạng sau: Dòng đầu mỗi test là hai số nguyên n ($2 \leq n \leq 10000$), và m ($0 \leq m \leq 50000$), tương ứng là số lượng học sinh khối 10 và số lượng mối quan hệ bạn bè còn lại sau chuyến đi chơi. Sau đó là n dòng, mỗi dòng ghi một số nguyên o ($-10000 \leq o \leq 10000$) thể hiện người đó nợ bao nhiêu (hoặc bị nợ nếu $o < 0$). Tổng n số nguyên này bằng 0. Kế đó là m dòng có dạng $x\ y$ ($0 \leq x < y \leq n-1$) thể hiện x và y còn là bạn bè.

Output

Với mỗi test, in ra một dòng POSSIBLE hoặc IMPOSSIBLE.

Ví dụ

Sample Input	Sample Output
2	POSSIBLE
5 3	IMPOSSIBLE
100	
-75	
-25	
-42	
42	
0 1	
1 2	
3 4	
4 2	
15	
20	
-10	
-25	
0 2	
1 3	

11966. ĐẶT TÊN TINH VÂN

Cao nằm trên sân thượng và ngắm sao. Bởi vì Cao không biết về bất kỳ tinh vân nào cả, nên Cao tự quyết định: nếu 2 ngôi sao đủ gần nhau – chúng sẽ thuộc về cùng một tinh vân. Và bây giờ, Cao sẽ đếm số lượng tinh vân trên bầu trời. Hai ngôi sao thuộc cùng một tinh vân nếu khoảng cách giữa chúng trên mặt phẳng hai chiều không vượt quá D đơn vị.

INPUT

Dòng đầu là số lượng test T ($T \leq 50$). Dòng đầu của mỗi test là 2 số: số lượng ngôi sao N ($0 \leq N \leq 1000$) và khoảng cách D là số thực ($0.00 \leq D \leq 1000.00$). N dòng kế tiếp là tọa độ của các ngôi sao, có dạng $X\ Y$ ($-1000.00 \leq X, Y \leq 1000.00$). Các số thực chỉ có nhiều nhất 2 chữ số sau dấu thập phân.

OUTPUT

Với mỗi test, in ra dòng "Case T: N". Ở đây T là số thứ tự test (tính từ 1) và N là số lượng tính vân.

Sample Input	Sample Output
2	Case 1: 2
5 1.5	Case 2: 3
1.0 0.1	
2.0 0.0	
5.0 0.2	
6.0 0.4	
3.0 -0.1	
3 4.0	
121.12 254.06	
645.04 301.85	
912.49 568.96	