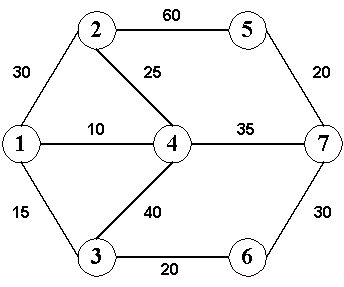
**Problem: THE TOURIST GUIDE - 10099 (10099.\*)**

Mr.G làm việc cho một công ty du lịch. Nhiệm vụ của anh ta là dẫn đoàn du lịch đến các địa điểm trong thành phố. Thành phố kết nối với nhau bằng một hệ thống đường 2 chiều. Mỗi con đường được cung cấp 1 số lượng hành khách tối đa có thể đi qua tùy theo chất lượng đường để tránh hư hỏng.Mr. G. có bản đồ thành phố cùng thông tin về số lượng người tối đa lưu thông trên đường. Anh ta hiểu rằng không phải lúc nào cũng có thể đưa tất cả hành khách đi chung 1 chuyến. Ví dụ thành phố gồm 7 điểm dừng như sau. Các cạnh biểu thị đường and và số ghi trên mỗi cạnh chính là lượng người tối đa được lưu thông trên đường đó trên 1 xe



Ví dụ anh ta muốn chở 99 hành khách từ điểm 1 --> điểm 7, Anh ta cần ít nhất 5 chuyến, tuyến đường anh ta sẽ đi là 1 - 2 - 4 - 7. , mỗi lần chở tối đa 24 khách + thêm anh tài xế nữa là 25

Nhưng MrG nhìn chung cũng chỉ là 1 lái xe , anh ta không phải lúc nào cũng đủ minh mẫn để tìm được đường tốt nhất.Bạn hãy giúp anh ta 1 tay

**Input “10099.inp”**

Input có thể có nhiều hơn 1 test case . Dòng đầu tiên gồm 2 số nguyên : *N* (N<= 100) and *R* số thành phố and số lượng đường. R dòng tiếp theo mỗi dòng gồm 3 số nguyên: *C*1, *C2* và *P*. *C*1 và *C2* là số hiệu của thành phố and *P* (P> 1) là số lượng người tối đa được phép lưu thông trên xe tại tuyến đường này. Số hiệu thành phố luôn từ 1 đến *N*. Dóng cuối cùng của test case gồm 3 số *S*, *D* và *T* biểu diễn điểm xuất phát , điểm đến và số lượng hành khách cần chuyển

**Input kết thúc bởi *N* và *R có giá trị 0*.**

**Output “10099.out”**

Mỗi test in ra số hiệu test case và số lần di chuyển tối thiểu để chở đủ số lượng hành khách theo yêu cầu.

Sau mỗi test là 1 dòng trắng

**Sample I/O**

|  |  |
| --- | --- |
| **10099.inp** | **10099.out** |
| 7 10 1 2 30 1 3 15 1 4 10 2 4 25 2 5 60 3 4 40 3 6 20 4 7 35 5 7 20 6 7 30 1 7 99 0 0 | Scenario #1 Minimum Number of Trips = 5 |

**Problem 6: HEAVY CARGO - 544 (544.\*)**

*Big Johnsson Trucks Inc.* là 1 công ty chuyên chở hàng hóa lớn. Công nghệ mới nhất của họ, the *Godzilla V12*, có thể chở hàng hóa dưới bất cứ hình dạng nào. Nó chỉ giới hạn bởi trọng tải của hàng hóa có thể làm hỏng con đường mà chiếc xe đi qua.

Cho thành phố xuất phát và đích đến, your công việc của bạn là xây dựng trọng tải tối đa cho chiếc xe *Godzilla V12* thấp nhất có thể để tiết kiệm chi phí cho nhà đầu tư mà vẫn chở được hàng được đến đích.

**Input  “544.inp”**

Input gồm nhiều test. Dòng đầu tiên gồm 2 số là số lượng thành phố *n* (2<=n<200) and số lượng đường *r* (1<=r<=19900) tạo nên mạng lưới giao thông.

Sau đó là r dòng gồm tên của 2 thành phố cùng với trọng tải tối đa cho phép của con đường này. Tên thành phố không dài quá 30 kí tự và không bao gồm kí tự trắng. Trọng tải giới hạn từ 0 - 10000. Đường luôn là đường 2 chiều.

dòng cuối cùng sẽ gồm 2 cái tên: thành phố xuất phát and thành phố giao hàng.

Input kết thúc với 2 số 0 cho n và r.

**Output  “544.out”**

Mỗi test case in ra 3 dòng:

* ``Scenario #*x*" x là số test case
* ``*y* tons" là số lượng trọng tải cần tìm
* 1 dòng trắng

**Sample I/O**

|  |  |
| --- | --- |
| **544.inp** | **544.out** |
| 4 3  Karlsruhe Stuttgart 100  Stuttgart Ulm 80  Ulm Muenchen 120  Karlsruhe Muenchen  5 5  Karlsruhe Stuttgart 100  Stuttgart Ulm 80  Ulm Muenchen 120  Karlsruhe Hamburg 220  Hamburg Muenchen 170  Muenchen Karlsruhe  0 0 | Scenario #1  80 tons  Scenario #2  170 tons |

**Problem: LONGEST PATHS 5.0 – 10000 (10000.\*)**

Ở một thế giới nọ, có 1 thằng giở hơi luôn muốn tìm đường xa nhất , dài nhất để đi ( tất nhiên là nhà nó trồng cây xăng ) để du lịch được nhiều nhất có thể. Điều buồn cười là nhà nó rất là giàu và nó không tiếc tiền để thực hiện điều mà nó thích.

Nó mới vung tiền ra để thuê bạn , 1 ông lập trình viên thông minh mà nghèo kiết xác. Ngu đâu mà bỏ lỡ cơ hội, bạn hãy viết 1 chương trình tìm ra con đường dài nhất từ 1 điểm cho trước ( là nhà của thằng ngu đó ) và điểm đến sau cùng của con đường ( để nó còn book máy bay đi về )

**Input “10000.inp”**

Input gồm nhiều test case. Dòng đầu tiên gồm 1 số nguyên dương *n* (1 < n <= 100) số lượng thành phố mà thằng điên có thể đi qua (i.e., số lượng node của graph).

Giá trị *n* = 0 kết thúc input.

Sau đó là số s , là nơi thằng điên sẽ bắt đầu xuất phát (1<= s<=n). Sau đó là các cặp số p q đánh dấu các đường kết nối , tức là sau khi đến thành phố p nó sẽ đi tiếp đến thành phố q

cặp 2 số 0 0 đánh dấu kết thúc test case.

An tâm rằng không có vòng lặp vô hạn trong cái graph ngu độn này

**Output  “10000.out”**

Với mỗi test case , bạn phải tìm ra con đường dài nhất từ nhà nó và điểm kết thúc chặng đường.In ra theo mẫu sau  
In ra 1 dòng trắng sau mỗi test case.

**Sample I/O**

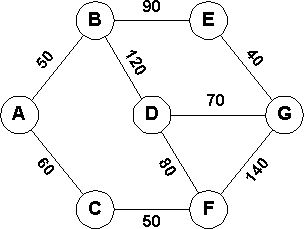
|  |  |
| --- | --- |
| **10000.inp** | **10000.out** |
| 2  1  1 2  0 0  5  3  1 2  3 5  3 1  2 4  4 5  0 0  5  5  5 1  5 2  5 3  5 4  4 1  4 2  0 0  0 | Case 1: The longest path from 1 has length 1, finishing at 2.  Case 2: The longest path from 3 has length 4, finishing at 5.  Case 3: The longest path from 5 has length 2, finishing at 1. |

**Problem: AUDIOPHOBIA - 10048 (10048.\*)**

Ô nhiễm tiếng ồn đang là một vấn đề rất phổ biến trên thế giới

Đơn vị đo âm thanh được sử dụng là *decibels* and âm thanh có độ lớn >= 130 decibels được cho là ô nhiễm và gây hại cho người nghe. Cường độ âm thanh trong tầm 60-­65 decibels được cho là bình thường..

Graph được cho dưới đây biểu thị các địa điểm và tiếng ồn tại các con đường giao nhau



Muốn đi từ A đến G thì có 1 con đường: **A­C­F­G**. Trong trường hợp này bạn phải nghe âm thanh là 140 decibels trên con đường từ F->G. Còn các đường **A­B­E­G**, **A­B­D­G** and **A­C­F­D­G** Thì âm thanh cao nhất bạn phải chịu đựng là 90, 120 và 80 decibels. Ngoài ra còn rất nhiều đường khác. Tuy nhiên, đường **A­C­F­D­G** là con đường phù hợp nhất và bạn không phải chịu bất cứ âm thanh nào vượt quá 80 decibels.

Trong problem này, bạn cần tìm ra âm thanh ô nhiễm nhất mà bạn phải chịu đựng khi di chuyển từ 2 điểm cho trước.

**Input  “10048.inp”**

Input gồm rất nhiều test case.

Dòng đầu tiên gồm 3 số C(<=100, S(<=1000) and Q(<=10000) C là số địa điểm, *S là số đường* và Q là số query.

S dòng tiếp theo có cấu trúc gồm 3 số: *c*1, *c*2 và *d* biểu thị âm thanh trung bình tại con đường đi từ *c*1 đến *c*2.

Q dòng tiếp theo gồm 2 số *c*1 and *c*2 (*c1*≠*c2*) để hỏi xem bạn cần chuẩn bị cái bịt tai có độ cản tối thiểu bao nhiêu decibels mới có thể đi từ *c*1 đến *c*2.

Input kết thúc bởi 3 số 0 cho *C*, *S* và *Q*.

**Output “10048.out”**

Bắt đầu mỗi test case bạn hãy in ra số thứ tự của nó (bắt đầu từ 1format trông giống output mẫu. Mỗi query in đáp án trên 1 dòng hiển thị cường độ âm thanh cần tìm (in decibels) mà bạn cần chịu đựng khi đi từ điểm c1 đến c2. Nếu không có đường kết nối giữa 2 điểm trên thì in ra ``no path".

In 1 dòng trắng giữa các test cases.

**Sample I/O**

|  |  |
| --- | --- |
| **10048.inp** | **10048.out** |
| 7 9 3  1 2 50  1 3 60  2 4 120  2 5 90  3 6 50  4 6 80  4 7 70  5 7 40  6 7 140  1 7  2 6  6 2  7 6 3  1 2 50  1 3 60  2 4 120  3 6 50  4 6 80  5 7 40  7 5  1 7  2 4  0 0 0 | Case #1  80  60  60  Case #2  40  no path  80 |

**Problem: 05-2 Rendezvous (11015.\*)**

Để phục vụ cho việc học nhóm,chúng tôi quyết định tập trung tại nhà của 1 thành viên. Team có tối đa 22 thành viên.

Nhưng chúng tôi lại gặp rất nhiều khó khăn trong việc chọn "căn cứ" cho nhóm mình, vì thế

chúng tôi quyết định gặp nhau tại địa điểm có tổng chi phí thấp nhất. Tổng chi phí thấp nhất được định nghĩa là tổng quãng đường từ căn cứ đến nhà của mỗi thành viên. Bài này bạn được thuê để xác định xem đâu là ngôi nhà phù hợp nhất để làm "căn cứ".

**Input  “11015.inp”**

Input chứa nhiều nhất 105 test case.Mỗi test case có cấu trúc như sau

Đầu tiên là 2 số N (1 ≤ N ≤ 22) và M (1 ≤ M ≤ (N ∧2−N)/2),

số lượng thành viên và số lượng đường. Sau đó là N dòng hiện thị tên của N thành viên có tối đa 10 kí tụ thường.

Sau đó là M dòng gồm 3 số i, j (1 ≤ i, j ≤ N) and k (1 ≤ k ≤ 1000). Có nghĩa là con đường từ i đến j dài k cây số, đường này là đường 2 chiều

Input N = 0 kết thúc file input

Số hiệu thành viên được đếm từ 1

**Output  “11015.out”**

Với mỗi test case bạn hãy in ra như sau:

“Case #i : XXX”

i thay bằng số test case và XXX thay bằng tên nhà thành viên vinh dự được chọn làm "căn cứ" với tổng đường đi nhỏ nhất. Nếu có nhiều hơn 1 thành viên phù hợp, thì in ra thành viên có số ID thấp hơn.

Note: Nhập tên tương ứng với số ID

số 1 : timotius

số 2 : harry

....

**Sample I/O**

|  |  |
| --- | --- |
| **11015.inp** | **11015.out** |
| 4 3  timotius  harry  richard  januar  1 2 10  1 3 8  1 4 6  4 3  rocky  herwin  gaston  jefry  1 2 5  1 3 5  1 4 5  0 0 | Case #1 : timotius  Case #2 : rocky |