

01. BFS

UVA 321. BIỆT THỰ MỚI

Mr.Black mới mua một ngôi nhà ở ngoại ô. Chỉ có một điều duy nhất làm ông không thoải mái: tuy mọi phòng đều có công tắc đèn nhưng bóng đèn mà công tắc điều khiển không nằm ở trong cùng phòng với công tắc đó. Dù người giúp việc của ông nghĩ đây là một ý kiến hay, ông cảm thấy những người thợ điện có vẻ ngốc khi họ nối dây điện giữa công tắc với bóng đèn. Một đêm, Mr.Black về nhà muộn. Khi đứng ở hành lang, ông nhận ra đèn ở mọi phòng đều tắt. Thật không may, Mr.Black sợ bóng tối, vì vậy ông không bao giờ vào một căn phòng không bật đèn. Sau một lúc suy nghĩ, Mr.Black quyết định tận dụng hệ thống bóng đèn. Ông cần phải đi đến phòng ngủ của mình và tắt đèn ở mọi căn phòng khác. Nhiệm vụ của bạn là phải viết chương trình, cho trước thông tin về căn nhà, xác định cách đi từ hành lang vào phòng ngủ khi ban đầu chỉ có đèn ở hành lang bật. Bạn không bao giờ được vào một căn phòng tối, và khi cuối cùng bạn vào được phòng ngủ thì đèn ở mọi căn phòng khác phải tắt. Nếu có nhiều đường tới phòng ngủ, bạn phải tìm ra đường đi với số bước bé nhất. Di chuyển từ một phòng này sang phòng khác, bật công tắc, tắt công tắc được coi là một bước.

INPUT

Input gồm nhiều test. Mỗi test bắt đầu với 1 dòng chứa 3 số nguyên r, d, s . r là số phòng của căn nhà, tối đa là 10. d là số cánh cửa nối giữa các phòng và s là số công tắc của căn nhà. Các căn phòng được đánh số từ 1 tới r . 1 là hành lang, và r là phòng ngủ. Tiếp đó là d dòng, mỗi dòng chứa 2 số i, j thể hiện có cửa từ phòng i tới phòng j . Sau đó là s dòng chứa 2 số k và l , thể hiện công tắc trong phòng k điều khiển bóng đèn trong phòng l . Giữa các test sẽ có một dòng trống. Input kết thúc khi $r=d=s=0$. Trường hợp này không cần xử lý.

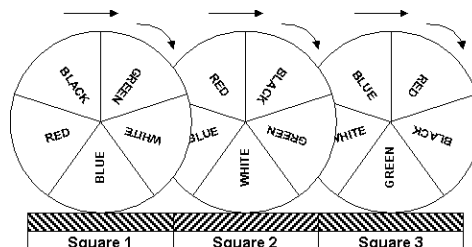
OUTPUT

Với mỗi căn nhà, bắt đầu với số thự tự của test: "Villa #1", "Villa #2",... Nếu như có phương án cho căn nhà này, in ra số bước ít nhất cần để đi tới phòng ngủ và cách đi. Nếu không có phương án, in ra một dòng duy nhất "The problem cannot be solved." Output một dòng trống sau mỗi test.

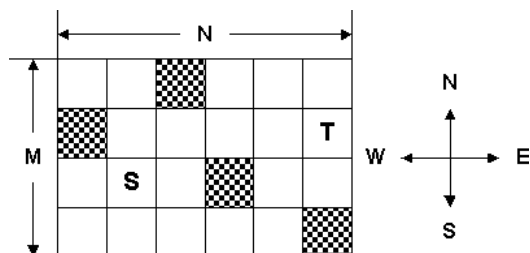
Sample Input	Sample Output
3 3 4	Villa #1
1 2	The problem can be solved in 6
1 3	steps:
3 2	- Switch on light in room 2.
1 2	- Switch on light in room 3.
1 3	- Move to room 2.
2 1	- Switch off light in room 1.
3 2	- Move to room 3.
	- Switch off light in room 2.
2 1 2	
2 1	Villa #2
1 1	The problem cannot be solved.
1 2	
0 0 0	

UVA10047. XE MỘT BÁNH

Monocycle là một chiếc xe đạp chạy bằng một bánh và chiếc xe của chúng ta sẽ có một chút đặc biệt. Bánh xe sẽ được tô màu theo 5 màu như hình bên. các phần được tô màu bằng nhau, chính xác 72o. Người lái xe đạp phải lái xe một bánh trên các ô vuông. Di chuyển từ tâm của 1 ô đến tâm của một ô tiếp theo khiến bánh xe quay chính xác 72o quanh trục của chính nó. Các hiệu ứng được biểu diễn như hình. Khi xe ở vị trí ô thứ 1, điểm giữa của phần màu



xanh chạm đất. Nhưng khi xe đi đến ô kế tiếp (square 2) thì điểm giữa của phần màu trắng chạm đất.



Một số ô có chứa vật chướng ngại và chặn xe không cho đi vào chỗ đó. Xe bắt đầu ở một vị trí nào đó và cố gắng đến vị trí mục tiêu trong thời gian ngắn nhất. Từ bất kỳ một ô nào, có thể đi thẳng đến ô tiếp theo hoặc đứng yên ở ô đó nhưng phải xoay 90o sang trái hoặc phải. Mỗi một hành động như vậy được tính là 1 giây.

Xe luôn bắt đầu quay mặt về phía Bắc và điểm chính giữa màu xanh lá hướng xuống mặt đất. Và tại ô mục tiêu, mặt màu xanh phải hướng xuống đất nhưng không cần quan tâm xe quay theo hướng nào. Trước khi xe chạy, bạn hãy tính xem thời gian tối thiểu để xe đi đến mục tiêu.

INPUT

Input có thể có nhiều test. Dòng đầu tiên của mỗi test là 2 số nguyên M và N, cho biết độ lớn của bản đồ. M dòng kế tiếp chứa N kí tự mỗi dòng. Kí tự '#' cho biết vị trí của một vật cản, phần còn lại xe được tự do di chuyển vào. Vị trí bắt đầu được đánh dấu bằng 'S' và mục tiêu được đánh dấu bằng 'T'. Input kết thúc khi M và N nhận giá trị 0.

OUTPUT

Mỗi test in ra một dòng chỉ thứ tự giống như trong Sample output. Nếu có thể đến được vị trí mục tiêu thì in ra thời gian ngắn nhất (theo giây) giống như trong Sample output, Nếu không thể, in ra "destination not reachable". In một dòng trống giữa 2 test.

Sample Input	Sample Output
1 3 S#T 10 10 #S.....# #..#.#.#. #.#.#.#.#. .#....#.#. ##.##..#.# #..#.#.#... #.....##. ..##.##... #.#.#.#... #.....##T 0 0	Case #1 destination not reachable Case #2 minimum time = 49 sec

UVA11101. ĐẠI SIÊU THỊ

Waterloo có hai siêu trung tâm mua sắm, mỗi cái trải rộng trên nhiều khu dân cư. Kim và Pat muốn mua sắm và đi bộ trong các trung tâm nhưng không muốn đi bộ giữa các trung tâm vì như thế không làm đúng nhiệm vụ: mua sắm. Bởi vậy, họ muốn biết khoảng cách nhỏ nhất giữa các trung tâm mua sắm.

Mỗi khu dân cư là một hình vuông đơn vị có ranh giới là các con phố và đại lộ. Các con phố chạy từ đông sang tây và đại lộ chạy từ bắc đến nam. Cả hai được đặt tên bằng các số nguyên liên tiếp nằm trong khoảng từ $0 \rightarrow 2000$ (đại lộ số nhỏ nằm phía tây đại lộ số cao và con phố số nhỏ nằm phía nam của con phố số cao). Các con phố và đại lộ có độ rộng bằng 0.

Mỗi trung tâm mua sắm là một tập hợp khu dân cư thông nhau. Ta định nghĩa hai khu dân cư thông với nhau qua một dãy các khu dân cư nếu hai khu dân cư liền kề có chung một cạnh. Các trung tâm mua sắm không giao nhau và không bao quanh các khu trống. Các khu trống không nằm trong các trung tâm thì thông nhau.

Input chứa nhiều test. Mỗi test mô tả hai trung tâm mua sắm. Mỗi trung tâm được thể hiện bằng số nguyên $p \geq 4$, chu vi của trung tâm, theo sau bằng một hoặc nhiều dòng chứa p cặp (a, s) là tọa độ các giao lộ nằm trên chu vi, theo chiều kim đồng hồ. Kết thúc bằng test chứa số 0.

Với mỗi test, in ra một số nguyên d – giá trị quãng đường đi bộ nhỏ nhất giữa các trung tâm mua sắm, giả sử rằng Kim và Pat luôn luôn đi bộ dọc theo các con phố và các đại lộ.

Sample Input	Sample Output
4 0 0 0 1 1 1 1 0 6 4 3 4 2 3 2 2 2 2 3 3 3 0	2

02. Dijkstra

UVA10801. THANG MÁY LÊN XUỐNG

Toà nhà chọc trời có không quá 100 tầng được đánh số từ 0 đến 99. Nó có n ($1 \leq n \leq 5$) thang máy để di chuyển lên hoặc xuống bằng các tốc độ khác nhau. Mỗi thang máy i trong khoảng từ 1 đến n cần T_i ($1 \leq T_i \leq 100$) giây để đi giữa 2 tầng cạnh nhau (lên hoặc xuống). Thang máy không nhất thiết phải dừng lại ở bất kỳ tầng nào, Tệ hơn nữa là một thang máy không nhất thiết phải phải dừng lại ở mọi tầng. Bạn đang ở tầng 0 và bạn muốn đi lên tầng k bằng cách nhanh nhất có thể. Giả sử bạn không cần đợi thang máy đầu tiên để đi vào và để cho đơn giản, việc chuyển thang máy ở một tầng nào đó mất đúng một phút. Dĩ nhiên, cả hai chiếc thang máy phải dừng lại ở tầng đó. Bạn không được sử dụng cầu thang bộ. Không ai đi cùng thang máy với bạn, nên bạn có thể dừng lại nếu bạn muốn. Tính thời gian tối thiểu để đi từ tầng 0 đến tầng k . Đi qua tầng k khi vẫn ở trong thang máy không tính là đến được.

OUTPUT

Input sẽ chứa nhiều test. Trong mỗi test, dòng đầu tiên sẽ bắt đầu bằng 2 số nguyên n và k . Dòng tiếp theo chứa các số T_1, T_2, \dots, T_n . n dòng tiếp theo chứa danh sách các số nguyên được sắp xếp – dòng đầu tiên sẽ là danh sách các tầng thang máy thứ nhất dừng, tiếp theo sẽ là danh sách các tầng thang máy thứ 2 dừng...

INPUT

Với mỗi test case, in ra 1 số - thời gian để đi từ tầng 0 đến k. Nếu không thể, in ra "IMPOSSIBLE".

Sample Input	Sample Output
2 30	275
10 5	285
0 1 3 5 7 9 11 13 15 20 99	3920
4 13 15 19 20 25 30	IMPOSSIBLE
2 30	
10 1	
0 5 10 12 14 20 25 30	
2 4 6 8 10 12 14 22 25 28	
29	
3 50	
10 50 100	
0 10 30 40	
0 20 30	
0 20 50	
1 1	
2	
0 2 4 6 8 10	

Giải thích ví dụ đầu tiên, dùng thang máy đầu tiên để đến tầng 13 (130 giây), đợi 60 giây để chọn thang máy 2 để đi tới tầng 30 (85 giây), tổng cộng 275 giây. Trong ví dụ thứ 2, đi thang máy 1 lên tầng 10, chuyển sang thang máy 2 và lên tầng 25, quay trở lại thang máy 1 và lên tầng 30. Tổng thời gian là $10*10 + 60 + 15*1 + 60 + 5*10 = 285$ s. Trong ví dụ 3, dùng thang máy 1 lên tầng 30, dùng thang máy 2 xuống tầng 20 và dùng thang máy 3 lên tầng 50. Trong ví dụ 1, thang máy 1 không lên tầng 1.

UVA11367. ĐỒ ĐẦY BÌNH

Sau khi trải qua một chuyến nghỉ hè quanh Châu Âu, bạn nhận ra rằng giá xăng giữa các thành phố là khác nhau. Liệu có thể tiết kiệm tiền xăng nếu bạn khôn ngoan hơn không? Để giúp đỡ các du khách khác hoặc có thể là chính bạn trong lần sau, hãy viết một chương trình xác định đường đi rẻ nhất giữa các thành phố bằng cách đổ xăng trên đường. Coi như mỗi chiếc xe đi đơn vị khoảng cách tốn một đơn vị xăng, bình xăng ban đầu rỗng.

INPUT

Dòng đầu tiên là hai số nguyên $1 \leq n \leq 1000$ và $0 \leq m \leq 10000$ tương ứng là số thành phố và số đường đi. Sau đó là một dòng có n số nguyên dương $1 \leq p_i \leq 100$, trong đó p_i là giá xăng tại thành phố thứ i. Theo sau là m dòng với ba số nguyên $0 \leq u, v < n$ và $1 \leq d \leq 100$, cho biết có đường đi giữa u và v với độ dài d. Sau đó là một dòng $1 \leq q \leq 100$, số lượng các câu hỏi, q dòng tiếp theo với ba số nguyên $1 \leq c \leq 100$, s và e, trong đó c là dung tích bình xăng, s là thành phố xuất phát, e là thành phố kết thúc.

OUTPUT

Với mỗi câu hỏi, in ra giá tiền của đường đi rẻ nhất giữa thành phố s và e, sử dụng một chiếc xe với dung tích xăng đã cho, hoặc "impossible" nếu không có đường đi giữa s và e với chiếc xe đã cho.

Sample Input	Sample Output
5 5	170
10 10 20 12 13	impossible

0	1	9
0	2	8
1	2	1
1	3	11
2	3	7
2		
10	0	3
20	1	4

UVA11492. BABEL

John và Mary là 2 anh em, và họ vô cùng hăng hái với khóa học ngoại ngữ sắp tới. Mỗi người đều tham dự một vài khóa học. Sau đó, họ bàn luận với nhau về từ vựng, ngữ pháp và các nền văn hóa của các nước. Họ nhận ra rằng có một số từ được dùng trong nhiều loại ngôn ngữ khác nhau, kể cả một số từ được dùng với các ngôn ngữ khác nhau. Ví dụ, từ “Amigo” trong tiếng Bồ Đào Nha và Tây Ban Nha đều có cùng nghĩa trong khi đó từ “date” trong tiếng Anh có nghĩa là ngày tháng trên lịch, còn trong tiếng Pháp lại có nghĩa là hoa quả. Còn “red” trong tiếng Tây Ban Nha là mạng và tiếng Anh lại là màu đỏ.

Vô cùng vui sướng vì phát hiện này, hai anh em quyết định viết ra tất cả các từ được dùng trong nhiều loại ngôn ngữ mà họ có thể nghĩ ra được, phụ thuộc vào các cặp ngôn ngữ cho trước. John thách đố Mary, cho trước 1 cặp ngôn ngữ viết ra 1 dãy các từ sao cho từ đầu tiên của dãy thuộc ngôn ngữ thứ nhất và từ cuối cùng thuộc ngôn ngữ thứ 2. 2 từ liên kế nhau đều là từ vựng của cùng một ngôn ngữ. Ví dụ như 2 ngôn ngữ Bồ Đào Nha và Pháp, Mary có thể viết ra các từ sau "amigo actual date" (Portuguese/Spanish, Spanish/English, English/French).

John vô cùng bất ngờ khi thấy Mary dễ dàng giải quyết vấn đề. Vô cùng khó chịu với em gái mình John quyết định tăng độ phức tạp của vấn đề Mary phải tìm cách in ra độ dài xâu bé nhất thỏa mãn yêu cầu (xâu là tập hợp các từ, không tính các dấu cách phân biệt giữa các từ) và 2 từ liên tiếp phải có chữ cái đầu tiên khác nhau.

John đã làm một cuộc tìm kiếm trên Internet và liệt kê ra một loạt các từ và bắt đầu thách đố Mary. Vì có thể có rất nhiều cách giải nên John chỉ yêu cầu Mary đưa ra độ dài xâu nhỏ nhất. Bạn có thể giúp đỡ Mary được không?

INPUT

Gồm nhiều test. Dòng đầu tiên là số m ($1 \leq m \leq 2000$), biểu thị tổng số lượng từ John đưa ra. Dòng tiếp theo là 2 xâu O và D biểu thị 2 ngôn ngữ. m dòng tiếp theo gồm các xâu $i1, i2$, và p thể hiện từ xuất hiện trong 2 ngôn ngữ ($i1$ và $i2$ khác nhau). Mỗi xâu có độ dài từ 1 đến 50, và chỉ được viết bằng các kí tự thường không viết hoa. Mỗi cặp ngôn ngữ có thể có nhiều từ nhưng một từ không được phép xuất hiện nhiều lần trong một test. Số không kết thúc Input.

OUTPUT

Với mỗi test, chương trình của bạn phải in ra một số nguyên thể hiện độ dài xâu nhỏ nhất.

Sample Input	Sample Output
4 portugues frances ingles espanhol red espanhol portugues amigo frances ingles date espanhol ingles actual 4 portugues alemao ingles espanhol red	12 impossivel 5

<div>espanhol portugues amigo frances ingles date espanhol ingles actual 6 portugues frances ingles espanhol red espanhol portugues amigo frances ingles date frances espanhol la portugues ingles a espanhol ingles actual 0</div>	
---	--