Combate à Dengue

Depois que João descobriu que estava com dengue, ele ficou muito irritado. Como nos últimos dias ele não saiu de casa, o mosquito que o picou só podia ser de algum foco de dengue perto de sua casa. Foi então quando ele teve uma ideia.

Assim que estiver um pouco melhor, João irá acabar com todos os focos de mosquitos que existem por perto de sua casa. Para realizar essa tarefa ele conseguiu um mapa, que pode ser visto como um plano cartesiano, onde sua casa e cada foco possuem uma coordenada distinta. Como a dengue é uma doença que deixa o corpo bem debilitado, João necessita de sua ajuda nessa tarefa.

João gostaria de saber qual a distância total mínima que ele gastará para sair de sua casa, visitar todos os focos de dengue exatamente uma vez e voltar para casa. Você consegue ajudar João em sua missão?

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste terá um inteiro **N** (1 ≤ **N** ≤ 15),representando a quantidade de focos de mosquito no mapa. Segue uma linha contendo dois inteiros **X** e **Y** (−100 ≤ **X**, **Y** ≤ 100), representando a coordenada da casa de João. Em seguida terão **N** linhas, cada uma contendo dois inteiros **X** e **Y** (−100 ≤ **X**, **Y** ≤ 100), representando a coordenada de um foco de dengue. A entrada termina quando **N** = 0 e não deve ser processada.

Saída

Para cada caso de teste imprima a distância mínima que João percorrerá, com duas casas decimais.

Exemplo de Entrada	Exemplo de Saída
4 0 0 1 2 2 3 2 2 3 3 0	8.89

Programação de Viagem

Valentina iniciou um novo desafio neste ano: ela irá dar aulas de programação! Ela está muito empolgada com essa atividade, uma vez que programar é uma de suas maiores paixões. Estas aulas ocorrerão durante a tarde em escolas de diversas cidades da sua região e ela deverá se deslocar de carro.

Como Valentina está no 3º ano do curso de Ciência da Computação, ela deve voltar de seu trabalho a tempo de ir para a universidade. As aulas em que ela leciona terminam por volta das 17h30min, e as aulas da faculdade iniciam às 19h30min. Sendo assim, ela possui 2 horas para poder voltar para a sua cidade e chegar na universidade a tempo do início da sua aula.

Muito esperta e organizada, ela coletou informações quanto o tempo médio, em minutos, de viagem entre as cidades onde ela tem que lecionar. Ela montou uma lista, onde ela enumerou as cidades que ela devia visitar, sendo a sua cidade de origem sempre a número 1. O tempo médio fornecido é bidirecional, ou seja, se da cidade 1 para a cidade 2 o tempo médio é 20 minutos, da cidade 2 para a cidade 1 é o mesmo.

Percebendo que ela está muito atarefada organizando suas aulas, você se ofereceu para ajudá-la construindo um programa que descobre qual é a melhor rota para ela voltar para sua cidade e se ela se atrasará para sua aula na faculdade ou não. Você sabe que existe pelo menos um caminho que leva até cada cidade, e só haverá um único menor caminho possível.

Entrada

A entrada consiste em diversas casos de teste. A primeira linha de cada caso contém dois inteiros **C** ($1 \leq C \leq 15$) e **E** ($1 \leq E \leq 225$), que indicam a quantidade de cidades e estradas. As **E** linhas seguintes contém três inteiros C_1, C_2 e **T**, que identificam o tempo médio **T** de deslocamento entre as cidades C_1, C_2. Por fim, um inteiro **D** identifica a cidade em que Valentina se encontra no momento. Uma linha com "0 0" finaliza a entrada.

Saída

Se a viagem durar menos que 2 horas, você deve imprimir "Will not be late. Travel time - M - best way - C_1 $C_{...}$ C_N" ("Não irá atrasar. Tempo de viagem - M - melhor caminho - C...), onde **M** é o tempo de viagem e C_N são a sequência de cidades que montam o melhor caminho. Caso contrário, você deve imprimir "It will be L minutes late. Travel time - M - best way - C_1 $C_{...}$ C_N" (Irá se atrasar L minutos. Tempo de viagem - M - melhor caminho - C...), onde **L** são os minutos que Valentina irá se atrasar.

Exemplo de Entrada	Exemplo de Saída
4 4 1 2 20 2 3 15 2 4 10 3 4 60 4 5 6 1 4 120 1 3 60 2 3 30 2 5 33 3 4 50 4 5 20 5 0 0	Will not be late. Travel time - 30 - best way - 4 2 1 It will be 3 minutes late. Travel time - 123 - best way - 5 2 3 1

See World

Sob a orientação da Profª Graziela Tonin, os estudantes de Tópicos Engenharia de Software estão desenvolvendo um sistema para ajudar o *See World*, o novo parque temático da cidade, a alocar orcas em dois tanques gigantes. As orcas, em particular, possuem uma hierarquia social bastante complexa, de pelo menos 4 níveis, e o curioso é que orcas são capazes de vocalizar diferentes dialetos, dependendo do nível da hierarquia social pelo qual se relacionam com seus interlocutores. Uma das maiores dificuldades em manter orcas em cativeiro é que, se duas orcas são postas juntas num mesmo tanque mas falam nenhum dialeto em comum, elas eventualmente brigarão, por não se entenderem, e por serem oriundas de culturas totalmente diferentes. Às vezes elas podem se machucar gravemente ou até morrer. Assim, o *See World* deseja alocar suas orcas nos seus dois tanques de modo que, se duas orcas forem alocadas num mesmo tanque, seja garantido que elas partilhem de ao menos um dialeto.

Entrada

A primeira linha da entrada consiste de um inteiro **N** ($1 \leq \mathbf{N} \leq 10_3$), o qual representa o número de orcas do *See World*. A propósito, cada orca do *See World* é representada unicamente por um único código entre 1 e **N**. Cada uma das **N** linhas seguintes consiste de **N** inteiros, de modo que o **j**-ésimo inteiro da **i**-ésima linha ($1 \leq \mathbf{i}, \mathbf{j} \leq \mathbf{N}$) é 1 se as orcas de códigos **i** e **j** partilham de ao menos um dialeto ou 0 caso contrário.

Saída

Imprima uma linha contendo unicamente a expressão **Fail!** se não é possível alocar as orcas nos tanques como desejado ou a expressão **Bazinga!** se é possível.

Exemplos de Entrada	Exemplos de Saída
5 1 1 1 1 0 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0 1 0 1 1	Bazinga!
5 1 1 0 1 0 1 1 0 0 1 0 0 1 1 0 1 0 1 1 1 0 1 0 1 1	Fail!

Promessa de Campanha

Durante sua campanha eleitoral, o prefeito do município de Barro Bravo prometeu que, até o fim de seu mandato, os cidadãos conseguiriam se locomover entre os principais pontos do município sem passar por nenhum trecho de estrada de terra (quando assumiu o cargo, não era possível ir a lugar algum sem passar pelo barro...).

A primeira providência que tomou foi finalizar as diversas vias de ligação que haviam sido parcialmente construídas, mas não terminadas. Assim que concluiu esta etapa, já com o orçamento reduzido, o prefeito precisava determinar se a promessa já fora cumprida ou não, e caso não tem sido, quantas estradas ainda deveriam ser construídas para que a promessa se concretizasse.

Escreva, portanto, um programa que auxilie o prefeito a obter sua resposta.

Entrada

A entrada consiste em uma série de casos de teste. O número **T** (**T** ≤ 100) de casos de teste é indicado na primeira linha da entrada.

Cada caso de teste é composto por várias linhas. A primeira e a segunda linha do caso de teste contém, respectivamente, os valores **N** (1 ≤ **N** ≤ 100) e **M** (0 ≤ **M** ≤ **N**(**N** - 1)/2), onde **N** é o número de pontos principais da cidade e **M** o número de estradas já construídas. Os principais pontos da cidade são identificados sequencialmente por números inteiros, a partir do número um.

As **M** linhas seguintes contém pares de valores **X** e **Y** (1 ≤ **X, Y** ≤ **N**), que indicam que existe uma estrada que liga o ponto **X** ao ponto **Y**.

Saida

Para cada caso de teste deverá ser impressa ou a mensagem "Caso #**t**: ainda falta(m) **E** estrada(s)" ou a mensagem "Caso #**t**: a promessa foi cumprida", conforme for o caso, onde **t** é o número do caso de teste (cuja contagem tem início no número um) e **E** é o número mínimo de estradas que devem ser construídas para que a promessa seja cumprida.

Ao final de cada mensagem deve ser impressa uma quebra de linha.

Exemplos de Entrada	Exemplos de Saída
4 3 2 1 3 2 3 4 2 1 2 3 4 3 0 6 5 1 2 1 3 1 4 2 3 3 4	Caso #1: a promessa foi cumprida Caso #2: ainda falta(m) 1 estrada(s) Caso #3: ainda falta(m) 2 estrada(s) Caso #4: ainda falta(m) 2 estrada(s)