**Problema A: As Árverezes Somos Nozes**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Um dos trabalhos dos monitores do curso é de arrumar problemas no sistema e garantir que as soluções sejam julgadas corretamente. Certo dia havia algo de estranho com um dos problemas e eles ficaram arrumando o sistema. No mesmo momento o Coffee Break foi servido e os alunos num impulso de fome foram todos comer. Há alguns salgados e doces na comida, e uns são mais populares do que outros. Depois de resolver o problema no sistema os monitores foram ao Coffee Break, e para a surpresa deles só havia um salgado de brócolis, que convenhamos é horrível. Por este motivo eles se abdicaram de escrever um enunciado para este problema.

Dado uma árvore com raíz no vértice **1**, imprima os vértices em pré-ordem. A árvore não é binária! Se você esqueceu o que isto significa chame um monitor, só não espere ajuda dos que perderam as Carolinas (um dos doces mais populares do Coffee Break).

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (2 <= **N** <= 10^4), indicando o número de vértices na entrada. Cada uma das **N-1** linhas seguintes representa uma aresta. Há dois números em cada linha, que são os identificadores de dois vértices conectados pela aresta. É garantido que a entrada seja uma árvore conectada.

**Saída**

Imprima a árvore em pré-ordem, para cada caso imprima os vértices e um espaço depois dos números.

|  |  |
| --- | --- |
| **Entrada de Teste**  2  2 1  3  2 1  3 1  4  1 4  1 2  1 3 | **Saída de Teste**  1 2  1 2 3  1 4 2 3 |

**Problema B: Turismo Sem Fronteiras**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Antes de embarcar no programa Turismo, ergh quero dizer Ciências Sem Fronteiras, Fulano precisa acabar com seu último trabalho de Algoritmos, que é sobre árvores.

O trabalho é simples, dado uma árvore com raíz no vértice 1, imprimir:

* A altura máxima.
* O maior grau de um nó da árvore.

Fulano é preguiçoso e te pagou para fazer o trabalho (que coisa feia einh?), ajude-o! Se você tem dúvidas do que é cada um dos itens acima chame um monitor.

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (2 <= **N** <= 10^4), indicando o número de vértices na entrada. Cada uma das **N-1** linhas seguintes representa uma aresta. Há dois números em cada linha, que são os identificadores de dois vértices conectados pela aresta. É garantido que a entrada seja uma árvore conectada.

**Saída**

Imprima separados por um espaço simples os valores dos itens pedidos no trabalho.

|  |  |
| --- | --- |
| **Entrada de Teste**  2  2 1  3  2 1  3 1  4  1 4  1 2  1 3 | **Saída de Teste**  2 1  2 2  2 3 |

**Problema C: GPS**

*TL: 2 segundos (~50 casos)*

*ML: 128 MB*

Você decidiu deixar o GPS de lado e traçar sua própria rota entre uma cidade e outra. Foi decidido que você começará na cidade 1 e depois seguirá para a cidade 2 usando uma estrada direta, depois para a 3, depois para a 4 e assim por diante até chegar à cidade **N**. Ou seja, você vai fazer o caminho 1, 2, …, **N**.

Neste problema cada estrada terá também um valor, que é a distância entre as cidades conectadas por ela. Dado o mapa das cidades e estradas imprima o menor caminho (escolhendo as menores arestas que ligam diretamente os vértices) seguindo o caminho 1, 2, …, **N**. **Imprima -1 se não existir caminho!**

**Entrada**

Há vários testes. A primeira linha de cada caso de teste contém os números **N** (2 <= **N** <= 10^4) e **M** (1 <= 10^5), indicando o número de cidades e estradas respectivamente. Nas próximas **M** linhas haverá três números inteiros **a**, **b** (1 <= **a,b** <= N) e **c** (1 <= **c** <= 10^3), indicando que as cidades **a** e **b** estão conectadas por uma estrada de tamanho **c**.

**Saída**

Para cada caso de teste, imprima o menor caminho de 1 até **N** ou -1 caso não exista caminho.

|  |  |
| --- | --- |
| **Entrada de Teste**  2 2  1 2 3  1 2 4  2 2  2 1 4  1 1 3  4 4  1 2 10  3 2 10  4 3 10  4 1 1  2 2  1 1 10  2 2 10 | **Saída de Teste**  3  4  30  -1 |

**Problema D: Torre de Babel**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Há várias pessoas trabalhando para construir a Torre de Babel. Cada pessoa fala duas línguas. Duas pessoas só podem conversar entre si se elas falarem uma língua em comum.

Uma pessoa pode também usar alguém de tradutor, ou seja, se uma pessoa sabe Inglês e quer conversar com uma pessoa que saiba Alemão, então se existir uma terceira pessoa que saiba Inglês e Alemão esta pessoa poderá traduzir a conversa.

Isto se estende para mais de um tradutor, por exemplo, considere uma pessoa que sabe a língua A e B, uma segunda pessoa que sabe a B e C, uma terceira que sabe C e D, e uma quarta D e E. A primeira pessoa pode conversar com a quarta, pois pode usar a terceira para traduzir o que a quarta disse e pode usar a segunda para traduzir o que a terceira traduziu! Ou seja, tivemos uma tradução de línguas D->C->B.

Seu objetivo é, dado as línguas de cada pessoa, identificar o grupo de maior tamanho, onde todas as pessoas deste grupo conseguem se comunicar. Caso tenha dúvida sobre a descrição do problema chame um monitor.

**Entrada**

Há vários testes. A primeira linha de cada caso de teste contém um número **N** (1 <= **N** <= 10^3), indicando o número de pessoas. Nas próximas **N** linhas haverá dois números inteiros **a** e **b** (1 <= **a,b** <= N), indicando as línguas que uma pessoa fala (a primeira destas linhas é referente a primeira pessoa, e assim por diante..). É possível que uma pessoa fale duas línguas repetidas!

**Saída**

Para cada caso de teste, imprima o maior conjunto de pessoas que consegue se comunicar.

|  |  |
| --- | --- |
| **Entrada de Teste**  7  1 2  4 6  5 7  6 7  3 1  2 3  6 4  3  1 1  2 2  3 3  3  1 2  3 3  2 2 | **Saída de Teste**  4  1  2 |

**Problema E: Ócio de Sessa**

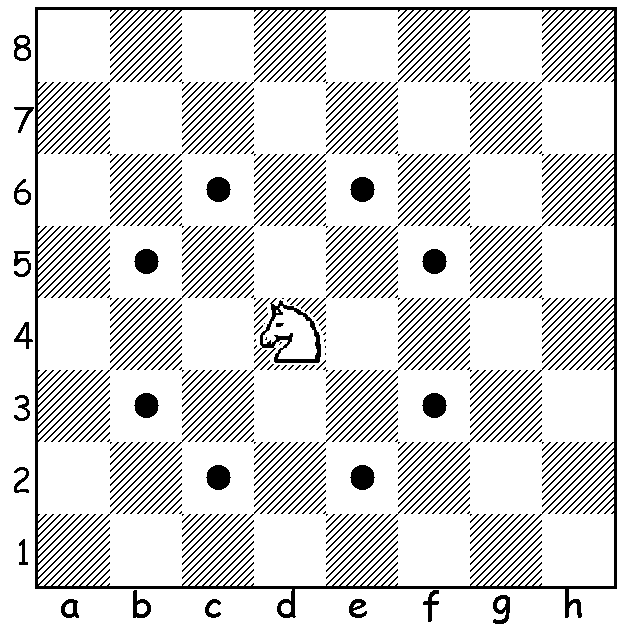
*TL: 1 segundo (~5000 casos)*

*ML: 128 MB*

Após ter perdido seu filho em batalha, o grande rajá de Taligana se encontrava em constante depressão e passou a descuidar-se de si e do reino. Ao saber de seu sofrimento, um jovem brâmane indiano chamado Lahur Sessa resolveu oferecer sua ajuda inventando um jogo de tabuleiro. Sessa explicou que a prática do jogo daria conforto espiritual ao rajá, que finalmente encontraria a cura para a sua depressão, o que realmente ocorreu. Esse jogo era o Xadrez.



Depois de ter salvo o reino, Sessa ficou sem nada pra fazer e passou a estudar seu próprio jogo. Certo dia, o jovem brilhante se deparou com a seguinte pergunta: Qual seria o menor número de movimentos com a peça do cavalo para conseguir move-la para qualquer casa do tabuleiro? O cavalo se movimenta em **L. (como mostra a figura abaixo)**



**Entrada**

Há varios casos de teste. Cada caso de teste é composto por uma linha contendo quatro inteiros (1 <= **Xa**, **Ya**, **Xb**, **Yb** <= 8) representando a posição na qual o cavalo se encontra (**Xa**, **Ya**) e a casa de destino ( **Xb**, **Yb** ). **ing**

**Saída**

Para cada caso imprima a quantidade mínima necessária de movimentos para andar com o cavado na posição (**Xa**, **Ya**) para a posição ( **Xb**, **Yb** ).

|  |  |
| --- | --- |
| **Entrada de Teste**  5 5 5 5  1 1 5 5  2 3 3 2  1 1 8 8 | **Saída de Teste**  0  4  2  6 |



**Problema F: Polonês da Polônia**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Jakub, o Polonês, está viajando pelo Brasil. Ele alugou uma Combi que tem **F** unidades de combustível. Ele tem em mãos um mapa das cidades e estradas. Viajar por uma estrada sempre gasta uma unidade de combustível.

Jakub pode começar sua viagem de qualquer cidade, e ele pode visitar uma cidade mais de uma vez. Ele quer saber quantas cidades diferentes no máximo ele consegue visitar com a quantidade de combustível disponível.

Há **N** cidades e exatamente **N-1** estradas. É sempre possível chegar de uma cidade a outra.

**Entrada**

A primeira linha de cada caso de teste contém os números **N** (1 <= **N** <= 10^4) e **F** (1 <= **F** <= 10^9), indicando o número de cidades e a quantidade de combustível. Cada uma das **N-1** linhas seguintes representa uma estrada. Há dois números em cada linha, que são os identificadores de duas cidades conectados pela cidade.

**Saída**

Imprima o número máximo de cidades que podem ser visitadas.

|  |  |
| --- | --- |
| **Entrada de Teste**  3 2  1 2  3 2  3 1  1 2  2 3  3 10  3 2  1 2 | **Saída de Teste**  3  2  3 |