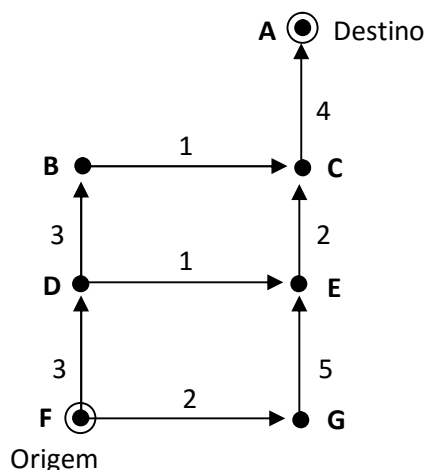


TRABALHO PARCIAL 01 - CONCEITOS FUNDAMENTAIS E CAMINHAMENTO

1. Dado um grafo valorado qualquer, determine o caminho mínimo entre dois vértices (origem e destino). Utilize o algoritmo de Dijkstra.



Exemplo de entrada

```
7
F A
0 I I I I I I
I 0 1 I I I I
4 I 0 I I I I
I 3 I 0 1 I I
I I 2 I 0 I I
I I I 3 I 0 2
I I I I 5 I 0
```

Saída para o exemplo de entrada

Caminho mínimo: F → D → E → C → A
Custo: 10

2. Dê uma solução para o problema Componentes Conexos (URI Online Judge | 1082) .
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1082>

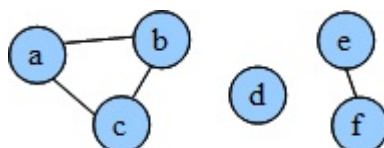
Com base nestas três definições:

Grafo conexo: Um grafo $G(V,A)$ é conexo se para cada par de nodos u e v existe um caminho entre u e v . Um grafo com apenas um componente é um grafo conexo.

Grafo desconexo: Um grafo $G(V,A)$ é desconexo se ele for formado por 2 ou mais componentes conexos.

Componente conexo: Componentes conexos de um grafo são os subgrafos conexos deste grafo.

O grafo a seguir possui 3 componentes conexos. O primeiro é formado pelos nodos **a,b,c**. O segundo é formado unicamente pelo nodo **d** e o terceiro componente é formado pelos nodos **e,f**.



Com base nestes conceitos, onde cada entrada fornecida que tem a identificação de cada um dos vértices, arestas e as ligações entre os vértices através destas arestas, liste cada um dos componentes conexos que existem no grafo, segundo a entrada fornecida.

Entrada

A primeira linha do arquivo de entrada contém um valor inteiro **N** que representa a quantidade de casos de teste que vem a seguir. Cada caso de teste contém dois valores **V** e **E** que são, respectivamente, a quantidade de Vértices e arestas (Edges) do grafo. Seguem **E** linhas na sequência, cada uma delas representando uma das arestas que ligam tais vértices. Cada vértice é representado por uma letra minúscula do alfabeto ('a'-'z'), ou seja, cada grafo pode ter no máximo 26 vértices. Cada grafo tem no mínimo 1 componente conexo.

Obs: Os vértices de cada caso de teste sempre iniciam no 'a'. Isso significa que um caso de teste que tem 3 vértices, tem obrigatoriamente os vértices 'a', 'b' e 'c'.

Saída

Para cada caso de teste da entrada, deve ser apresentada uma mensagem **Case #n:**, onde **n** indica o número do caso de teste (conforme exemplo abaixo). Segue a listagem dos vértices de cada segmento, um segmento por linha, separados por vírgula (inclusive com uma vírgula no final da linha). Finalizando o caso de teste, deve ser apresentada uma mensagem indicando a quantidade de componentes conexos do grafo (em inglês). Todo caso de teste deve ter uma linha em branco no final, inclusive o último caso de teste.

Obs: os nodos devem sempre ser apresentados em ordem crescente e se há caminho de a até b significa que há caminho de b até a.

Exemplo de entrada

```
3
3 1
a c
10 10
a b
a c
a g
b c
c g
e d
d f
h i
i j
j h
6 4
a b
b c
c a
e f
```

Saída para o exemplo de entrada

```
Case #1:
a,c,
b,
2 connected components

Case #2:
a,b,c,g,
d,e,f,
h,i,j,
3 connected components

Case #3:
a,b,c,
d,
e,f,
3 connected components
```

Observações:

1. o trabalho pode ser feito em dupla. A interpretação do enunciado faz parte da avaliação;
2. a avaliação será feita sobre os programas-fonte entregues ao professor;
3. serão consideradas a racionalidade e lógica da solução;
4. coloque seu nome como comentário no início de cada programa-fonte;
5. as soluções devem ser feitas OBRIGATORIAMENTE na linguagem Java. Os arquivos de entrada e saída devem ser gravados no diretório "c:\temp". Caso este item não seja respeitado, as respectivas questões serão ZERADAS.
6. os programas-fonte devem ser postados no AVA até o dia **24/05/2016**.