



Objetivos de la Guía

- Aprender sobre los grafos
- Aprender sobre la implementación matriz de adyacencia en C++
- Aprender sobre la implementación lista de adyacencia en C++

> A - Soy el mapa

El mapa de Dora la exploradora se va a retirar porque ya está muy viejo, así que busca entrenar a un nuevo mapa que sea el nuevo mapa. Como hoy en día, los mapas tienen el poder de los computadores, parte del entrenamiento consiste en realizar programar de C++ para poder leer y mostrar mapas.

* Parte 1: Mostrar la información de matriz

La primera parte del entrenamiento consiste en hacer uso de la matriz de adyacencia para poder representar un mapa. Su trabajo es realizar un código que lea un grafo por consola y muestre por cada nodo los vecinos a los que puede llegar.

* Input:

La primera línea consiste en dos enteros n y m , donde n corresponde a la cantidad de nodos y m corresponde a la cantidad de aristas del grafo respectivamente.

Luego, le siguen m líneas donde cada línea tiene dos enteros a y b ($1 \leq a, b \leq n$) correspondiente a que entre la ciudad a y b existe una arista bidireccional.

```
3 4
1 2
2 3
3 3
1 3
```

* Output:

Muestre por pantalla líneas, donde en la i -ésima línea se muestre todos los vecinos del nodo i . Si el nodo no tiene vecinos, la línea quedará vacía. No importa el orden en que se muestran los vecinos.

```
2 3
3
1 2 3
```

* Otros ejemplos:

Input

```
1 0
```

Output



Input

```
6 4
6 2
2 3
2 2
3 3
6 3
```

Output

```
2 3
3 2 6

3 2
```

Tip 1

Recuerde que en la matriz de adyacencia si en la posición (i,j) hay un 1 es porque existe la arista entre i y j , si hay un 0 es porque no existe.

* Parte 2: Ahora con lista de adyacencia

Realice el mismo problema anterior, pero ahora utilizando lista de adyacencia. Utilice los mismos ejemplos anteriormente planteados.

Tip 1

Recuerde que en la lista de la adyacencia, en la posición i del vector se encuentran los vecinos de i .



> B - El amigo del amigo

Mak Suqerver creó la red social OCIBook y tiene una pregunta, ¿Qué amigos tienen en común dos personas?. Para eso les pide ayuda a ustedes, en hacer un programa que reciba la cantidad de personas en su red social. Además, de recibir si es que una persona a es amiga de una persona b .

* Input:

La primera línea consiste de dos enteros n , dónde n corresponde a la cantidad de personas dentro de la red social.

Luego, le siguen $n \cdot n$ líneas, dónde en la i -línea se encuentra dos enteros a_i y b_i , correspondiente a que las personas a_i y b_i son amigas. Luego, le sigue una última línea que corresponde dos enteros x e y , dónde x e y son las personas que queremos saber cuáles son sus amigos en común.

```
6 4
1 2
2 3
1 3
4 3
1 3
```

* Output:

Muestre por pantalla todos los amigos que tienen en común x e y . Da lo mismo el orden en que muestres a los amigos en común.

```
2
```

* Otros ejemplos:

Input

```
8 5
3 2
4 5
3 5
4 6
6 3
4 3
```

Output

```
5 6
```

Tip 1

Para este problema imaginen que:

- las personas son nodos
- si a y b son amigos, entonces existe una arista entre a y b .