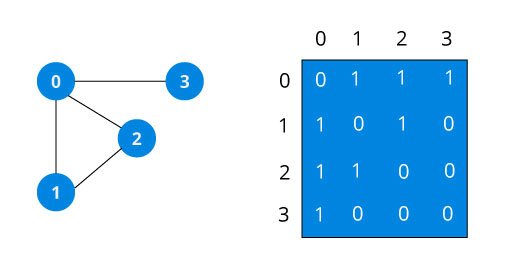
**OBJECTIVES:**

In this lab we learnt about;

* Implementing graph using adjacency matrix

**12.2.1 Adjacency Matrix** An adjacency matrix is a way of representing a graph G = {V, E} as a matrix of booleans.  
  
**12.2.2 Adjacency Matrix Representation** The size of the matrix is VxV where V is the number of vertices in the graph and the value of an entry Aij is either 1 or 0 depending on whether there is an edge from vertex i to vertex j.The image below shows a graph and its equivalent adjacency matrix.



In case of undirected graph, the matrix is symmetric about the diagonal because of every edge (i,j), there is also an edge (j,i).

**12.2.3 Adjacency Matrix Operations (undirected graph)**

**- void addEdge(int i, int j)**  set Aij to 1 (indicates that edge exists)  
**- void removeEdge (int i, int j)** set a[i][j] to 0 (indicates edge does not exist)  
**- bool isEdge (int I, int j)**  return Aij

**LAB TASKS:**

**TASK 1:**

Implement graphs with the following operations

a) Graph Creation

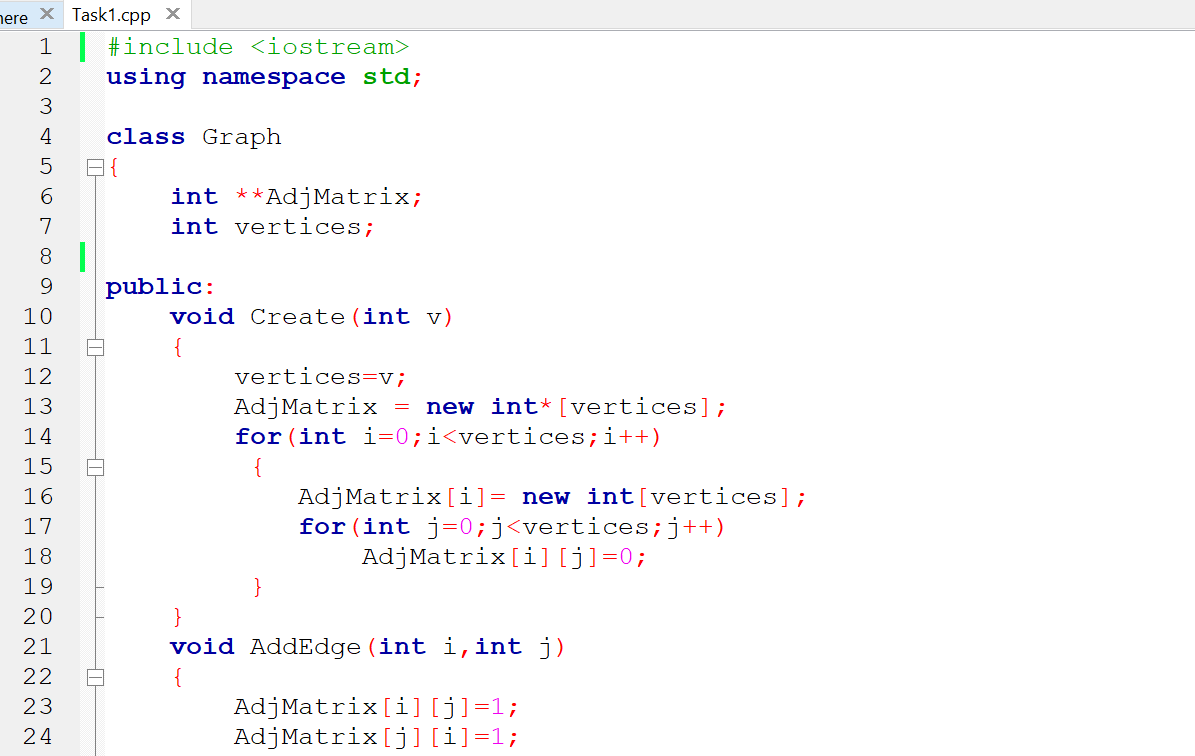
b) Adding Vertex to Graph

c) Removal of Vertex from the graph

d) Checking whether an edge exist between two vertices

e) Printing graph

**CODE:**

****

**A screen shot of a computer code

Description automatically generated with low confidence**

**A screenshot of a computer code

Description automatically generated with low confidence**

**A screen shot of a computer code

Description automatically generated with low confidence**

**A picture containing text, screenshot, software, number

Description automatically generated**

**OUTPUT:**

A screenshot of a computer program

Description automatically generated with medium confidence