Exercise 03 - Graphs

Create a C++ file named '`exercises03.cpp`' that only includes the libraries '`iostream`', '`string`', '`cstdlib`', and '`ctime`', and defines and tests the following functions.

1. Define a void function named `matrixview()` that takes a graph (a two-dimensional Boolean array) and its number of vertices (an unsigned integer) as parameters and displays the graph as a labeled adjacent matrix with an O indicating an edge between vertices and an X otherwise. An adjacent matrix is an $n \times n$ Boolean matrix where n is the number of vertices that specifies which pair of vertices has an edge between them.

   Example: The invocation of `matrixview({{false, true, true}, {true, false, false},{false, true, false}}, 3)` will display

   ```
      1  2  3
   1  X  O  O
   2  O  X  X
   3  X  O  X
   ```

2. Define a Boolean function named `undirectedgraph()` that takes a graph (a two-dimensional Boolean array) and its number of vertices (an unsigned integer) as parameters and returns true only if the graph is not a digraph; otherwise, it returns false. The adjacent matrix of an undirected graph is symmetric.

3. Define a void function named `degrees()` that takes a graph (a two-dimensional Boolean array) and its number of vertices (an unsigned integer) as parameters and displays the degree of each vertex on separate lines.

   Example: The invocation of `degrees({{false, true, true}, {true, false, false},{false, true, false}}, 3)` will display

   ```
   deg(1) = 3
   deg(2) = 3
   deg(3) = 2
   ```