The Tasks

- In this assignment, the task is to implement a graph with adjacency-list representation, and implement connected component labeling (equivalent to finding equivalent classes in chp. 4) using either DFS or BFS.
- Name your graph class **`MyGraph`**.
- For the input, the graph is initialized with a binary array, which represents an adjacency matrix. You need to convert it to adjacency-list representation in your class.
  - The constructor of the graph class take two inputs: The number of vertices, **`N`**, and a 1-D array (type **`int`**) of size **`N*N`** for the adjacency matrix.
  - You can assume that the input matrix is symmetric.
- The graph vertices are indexed from **`0`** to **`N-1`**.

The Tasks

- Example construction:

```
int x[] = {0,0,1,0,0,0,0,1,1,0,0,0,0,1,0,0};
MyGraph G(4, x); // a graph with 4 vertices
```

- Output the connected components you find with text. Show the vertices in each connected component in one line. Example output for the above graph:

```
Component#1: 0 2
Component#2: 1 3
```

- Implement either BFS or DFS as a member function. It takes one input that is the starting vertex index.
- Implement a member function **`CC`** to find the connected components and do the text output using your **`BFS`** or **`DFS`** function. This function takes no input.

The Guidelines

- Allowed environments: VS2012/2013/2015, Dev-C++. Indicate your environment at the beginning of your code.
- You need to write your own **`main`** function to test your permutation generation function. You do not need to include this **`main`** function in your submission. The instructor will provide a test **`main`** function for you.
- You can use any of these STL class templates: **stack**, **deque**, and **list**, for this assignment.
- Include documentation; this will be part of your grade.
- Demo: Only a randomly selected subset of students; will be announced separately after the due date.

The Guidelines

- Submission:
  - Use E3 only.
  - Submit all your code in a single header file (`.h`). Name it `P5_xxxxxx.h`, where `xxxxxx` is your ID. **Do not** submit your `main` function or any file that is not your code (such as the *.sln file). No compressed file (*.zip, *.rar, etc.). **Only the header file!!!**
  - Due date: **1/1/2016**. There's a grace period of 4 days with 10% deduction per day. (The deduction kicks in only when you have accumulated more than three days of delay during the semester.)