A template with a preliminary implementation is provided in BeachBoard (template.zip) under *Template* of the *Content Tab*. The template provides the classes and a simple menu to interact. Modify the menu to support the required functionalities. Read the readme.txt file for the documentation.
Note Only assignments that use the template will be graded.

LAB 7: GRAPHS

Learning objectives: CLO 1, CLO 3, CLO 4, CLO 5
Use Python 3.8 or higher for the assignment:

1. Implement AdjacencyMatrix and AdjacencyList including $add\_edge(i, j)$, $remove\_edge(i, j)$, $has\_edge(i, j)$, $out\_edges(i)$, $in\_edges(j)$ and the traversal $BFS(i)$ and $DFS(i)$.

   Learning objectives: CLO 1, CLO 3

   Test your program:

   - Remove from an empty graph, e.g., $remove\_edge(3, 5)$.
   - Check if an edge does exists, e.g., $has\_edge(3, 5)$.
   - Add edges $(1, 2), (2, 3), (3, 4), (4, 1), (1, 3)$, $add\_edge(1, 2)$, $add\_edge(2, 3)$, $add\_edge(3, 4)$, $add\_edge(4, 1)$, $add\_edge(1, 3)$.
   - Check for the edge $(1, 2)$ and $(1, 2)$. The first one must exist and the second must not exist.
   - Compute the in edges of 3, i.e., $in\_edges(3)$. It should return $(2, 3), (1, 3)$ in any order.
   - Compute the out edges of 1, i.e., $out\_edges(1)$. It should return $(1, 2), (1, 3)$ in any order.
   - Display the output of $BFS(1)$. The output must be $1, 3, 2, 4$ or $1, 2, 3, 4$.
   - Display the output of $DFS(1)$. The output must be $1, 2, 3, 4$ or $1, 3, 4, 2$.

2. Book Store System. Recall that books.txt contains thousand of books and dvd titles. Each title corresponds to a single row with five register separated by ^ as follows:

   $key\hat{}title\hat{}category\hat{}rank\hat{}similar$.

   Moreover $similar$ consists of the number of entries that are similar to it and then the keys to the similar books. For example,

   ```
   0827229534^Patterns of Preaching: A Sermon Sampler^Book^396585^5  0804215715  156101074X  0687023955  0687074231  082721619X
   ```

   has 5 books that are similar and their keys are $0804215715, 156101074X, 0687023955, 0687074231, 082721619X$.

   Learning objectives: CLO 1, CLO 3, CLO5

   (a) Load all the books in an instance $bookCatalog$ of your the ArrayList as in lab1 and in an instance $indexKeys$ of your ChainedHashTable where the key is the key if the book and the value is the index in $bookCatalog$, i.e.,

```
For each row i in books.txt
     b = Book(key, title, group, rank, similar)
     bookCatalog.append(b)
     indexKeys.add(key, i)
```

Then, create an instance $similarGraph$ of the AdjacencyList with n vertices and add the edges accordingly.

```
For each row i in books.txt
     key, title, group, rank, similar
     l = similar.split()
     For k in range(1, len(l)):
        j = indexKeys.find(l[k])
        if j is not None:
          similarGraph.add_edge(i, j)
```

(b) Display all the books that are similar up to k degree. Given a book index r, use the BFS in the graph to display the books that are at distance k from r. Modify the BFS to accept k that stops when all the nodes at distance at most k has been visited. Add an option in the menu.

(c) Check if degree of separation of any two books. Given the book indexes r_1, r_2, use the DFS in the graph to display the length of the shortest path between r_1 and r_2 if it exists. Modify the DFS algorithm to stop when the shortest path is found. Add an option in the menu.

Test your program:

- Display the degree of separation between 0 and 159811: It should be 11427
- Display the degree of separation between 0 and 159810: It should be 1

Submit all the source code (Python files (.py) in a zip file. The name of the zip file with the source code must be your first name, second name, and the data structure separated by a hyphen. For example, oscar-ponce-graphs.zip.
Submissions that do not follow the previous specification will be rejected and you will have 0 in the lab.

RUBRICS

	Level 4 2 Pt	Level 3 1.5 Pt	Level 2 1 Pt	Level 1 0.5 Pt
AdjacencyMatrix and AdjacencyList implementation	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete
List all similar books at distance k	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete
Compute the degree of separation between any two nodes k	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete