CSIS 3475 – 002 (Fall 2023)

Assignment# 1 (10%)

Due: Friday Oct 20th, 2023 (End of the day)

Total Points: 30

==========

For this assignment create a Java project named **firstnameA1_3475_002**, where firstname is your first name. Save all the required Java classes/files in this project.

Submit your Java project on Blackboard. Maximum 3 attempts allowed. Individual student work; submit your own work.

In case of cheating - if two or more students submit identical or copied code, all students will receive a zero on the assignment. Any violations of the Academic Integrity policy will be reported to the appropriate administrator for adjudication.

Note: Don't forget to add comments including Javadoc comments to your code. Generate Javadoc for your project.

Part a: (User-defined linked list): (Points: 10)

Create an interface **ListInterface** with the following methods

a) add(newEntry): Adds a new entry to the end of the list.
b) add(newPosition, newEntry): Adds a new entry to the list at a given position.
c) remove(givenPosition): Removes the entry at a given position from the list.
d) clear(): Removes all entries from the list.
e) replace(givenPosition, newEntry): Replaces the entry at a given position in the list with a given entry.
f) getEntry(givenPosition): Retrieves the entry at a given position in the list.
g) toArray(): Retrieves all entries in the list in their current order.
h) contains(anEntry): Sees whether the list contains a given entry.
i) getLength(): Gets the number of entries in the list.
j) isEmpty(): Sees whether the list is empty.

Create a **LList** class (user-defined and generic) that implements ListInterface mentioned above. The program is to work with a List ADT using a linked chain approach.

Part b: (an application of LList class created in part a) (Points: 20)

In order to create your Social Media application, you will need to create a class named **SocialMediaApp** that utilizes a linked list class (created in part a) to store a person's information, including their name, email, location, and a list of their friends (list). For a better user experience, it is suggested to create a menu that includes the following options:

- Add a person to the list. Prompt user for the person's details including name, email, location and a list of friends.
- Add a person at a particular position to the list. Prompt user for the person's details including name, email, location and a list of friends.
- Remove a person.
- Remove all the persons from the list/clear the list.
- Retrieve and display the information of everyone in the list.
- Search a person by name or by email and display the person's details including a friend list.
- Add a friend to the friend list or remove a friend from the friend list after searching for a particular person.
- Count the number of people on the list.
- Check if the list is empty.
- Exit

The program should continuously display the menu until the user chooses to exit. For each option on the menu, execute the necessary operation and display the output.

Hints: Create a **Person** class with the following fields: name, email, location and a list of friends.

Submission:

- Compressed/zipped your **firstnameA1_3475_002** project and upload it on Blackboard.
- Three submissions are allowed, last one will be graded.