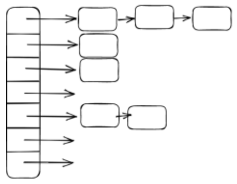
Data Structures Course Project - Fall 2022 [Banking System]

Important Rules:

- This project will be a teamwork project, with a team consisting of 4 or 5 members **FROM THE SAME LAB.**
- Take attention to handle all exception cases by the user throughout the project.
- All team members **MUST** understand everything in the code.
- Grades are distributed as follows: [Part #1 = 15 Points, Part #2 = 25 Points, Part #3 = 50 Points, Part #4 (Bonus Part) = 15 Points, Code Quality = 10 Points] => Total Points = 100 + 15 Bonus Points.
- Code quality is measured according to naming conventions, writing comments, following dynamic and static nature of program logic.
- Please follow the naming convention of methods, attributes, classes, etc.
- You'll submit a file with **members ids** and with **.cpp** extension.
    - [e.g., 20000000_20000000_20000000_20000000_20000000.cpp].
- Grades are based on the discussion.
- You will find ideas and helpful tips in the previous code tips of assignment 1 & 2.
- Cheating will be graded ZERO.

Part #1: Classes & Templates

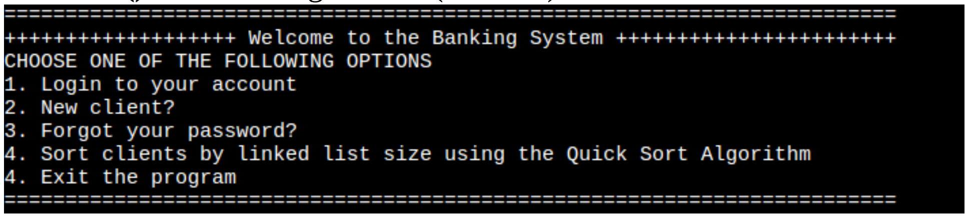
- **Client Class (5 Points)**
    - This class will contain the following attributes, string clientName, string clientEmail, string clientPhone, stringClientAddress, string clientPassword, int clientID, and array of type Transaction with size 5.
    - Implement all the setters and getters for the above attributes, implement a parameterized constructor to take the clientName, clientEmail, clientPhone, and clientAddress and set them to the class attributes.
    - Implement a method to print the client info, except the password and the list of transactions.
- **LinkedListNode Class (2 Point)**
    - This class is to represent the linked list node, it will be of type Client, the node will have two attributes, data part and next part, and one constructor to take the data and set it and next = NULL.
- **ClientsLinkedList Class (2 Points)**
    - This is the class representing the linked list of clients, with two attributes, head and listSize, and one constructor to set the head = NULL.
- **Transaction Class (3 Points)**
    - This class is to represent the transaction, with the attributes: int transactionID, string transactionName, string transactionDate (e.g., transactionDate = "14/Dec/2022").
    - Implement getters and setters for this class and the parameterized constructor, and a function to print the transaction info.
- **Array of Clients: (3 Points)**
    - This part you need to initialize an array of size 10, the array type is LinkedListNode because each index in the array will contain a pointer to a linked list.

- See this image for more information
  - 

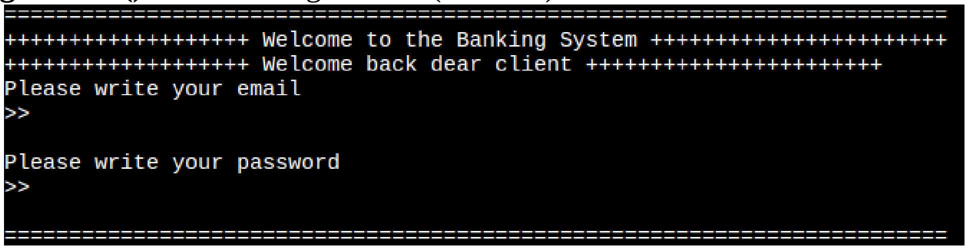
===

Part #2: Menus

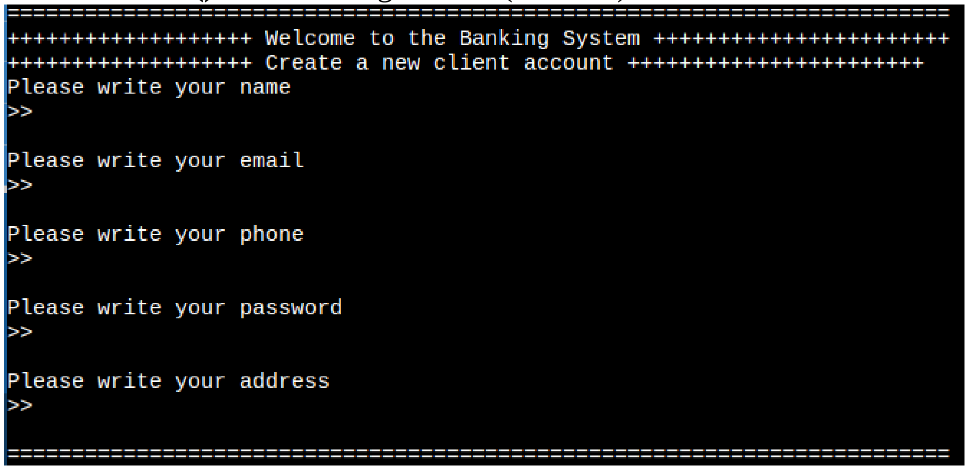
You need to create a method for each of the following menus:

- **void mainMenu():** see the image below (3 Points)
  - 

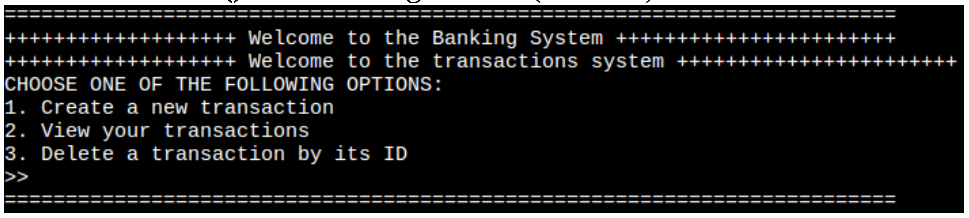
```
==========================================================================
+++++++++++++++++++ Welcome to the Banking System +++++++++++++++++++++++
CHOOSE ONE OF THE FOLLOWING OPTIONS
1. Login to your account
2. New client?
3. Forgot your password?
4. Sort clients by linked list size using the Quick Sort Algorithm
4. Exit the program
==========================================================================
```

- **void loginMenu():** see the image below (3 Points)
  - 

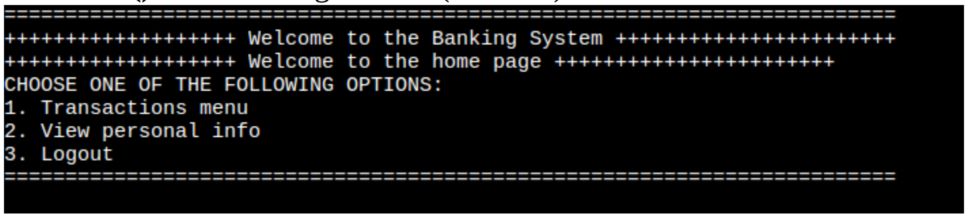
```
==========================================================================
+++++++++++++++++++ Welcome to the Banking System +++++++++++++++++++++++
+++++++++++++++++++ Welcome back dear client +++++++++++++++++++++++
Please write your email
>>

Please write your password
>>

==========================================================================
```

- **void newClientMenu():** see the image below (3 Points)
  - 

```
==========================================================================
+++++++++++++++++++ Welcome to the Banking System +++++++++++++++++++++++
+++++++++++++++++++ Create a new client account +++++++++++++++++++++++
Please write your name
>>

Please write your email
>>

Please write your phone
>>

Please write your password
>>

Please write your address
>>

==========================================================================
```

- **void transactionsMenu():** see the image below (3 Points)
  - 

```
==========================================================================
+++++++++++++++++++ Welcome to the Banking System +++++++++++++++++++++++
+++++++++++++++++++ Welcome to the transactions system +++++++++++++++++++++++
CHOOSE ONE OF THE FOLLOWING OPTIONS:
1. Create a new transaction
2. View your transactions
3. Delete a transaction by its ID
>>
==========================================================================
```

- **void clientHome():** see the image below (3 Points)
    - 

```
=========================================================================
++++++++++++++++++++ Welcome to the Banking System ++++++++++++++++++++++++
++++++++++++++++++++ Welcome to the home page ++++++++++++++++++++++++
CHOOSE ONE OF THE FOLLOWING OPTIONS:
1. Transactions menu
2. View personal info
3. Logout
=========================================================================
```

- The first menu should appear is the mainMenu(), user can login by email and password from the loginMenu(), or if the user is new, he/she can create a new account using the newClientMenu(), user can create a transaction through the transactionsMenu().
  (5 Points)
- You **must** handle all moves from one menu to another, user must be able to go back from any menu to the mainMenu(), you must handle exception cases. (3 Points)
- Forget password part, the user will write correct and exist email, then you will print the related password with this account. (2 Points)

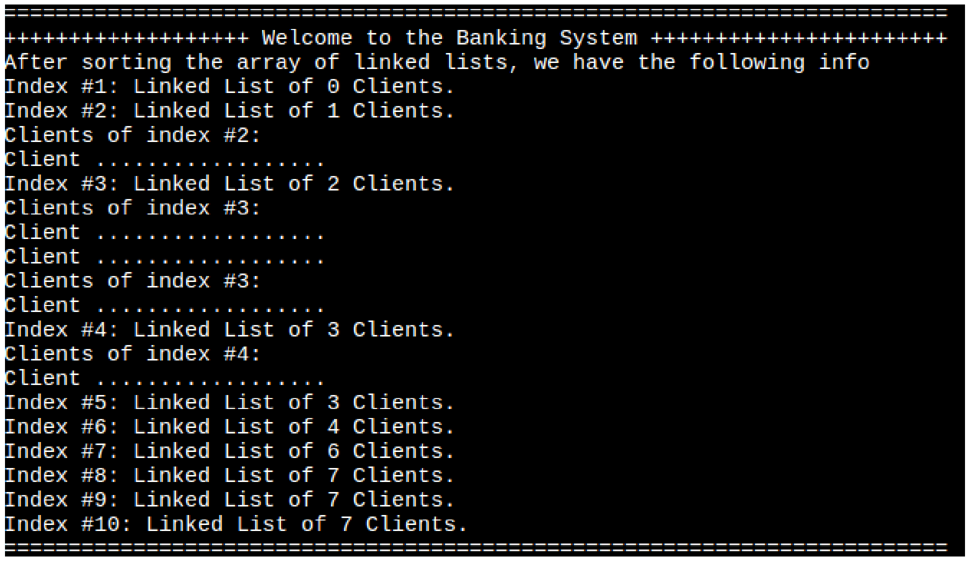
===

Part #3: Program Logic

A client will login to the system, by writing the correct email and password, then you will show the clientHome() menu, a client can go to the transactions page, view all personal info except the password, or log out. In log out, you will go back to the mainMenu().

You must handle an array of size 10 (Refer to part #1), to store the clients, each index of this array will point to a linked list of clients, while creating a new account, a new client will be added to the system, so after taking the entered information from the user – as showed in the newClientMenu() - in addition to clientID which will be generated automatically using a RANDOM function, you have to hash/assign this new client to a specific index of the array, so the client will be added to the end of the linked list in that array index. Using linked list here will help us avoid the collision problem, as more than one client might be mapped to the same array index. (25 Points)

As in the mainMenu() you have the option to sort the array of linked lists by the linked list size using the Quick Sort Algorithm, then view each array index after sorting and print the clients in the linked list. (See the image below for more info) (15 Points)



```
========================================================================
++++++++++++++++++ Welcome to the Banking System ++++++++++++++++++++++
After sorting the array of linked lists, we have the following info
Index #1: Linked List of 0 Clients.
Index #2: Linked List of 1 Clients.
Clients of index #2:
Client .................
Index #3: Linked List of 2 Clients.
Clients of index #3:
Client .................
Client .................
Clients of index #3:
Client .................
Index #4: Linked List of 3 Clients.
Clients of index #4:
Client .................
Index #5: Linked List of 3 Clients.
Index #6: Linked List of 4 Clients.
Index #7: Linked List of 6 Clients.
Index #8: Linked List of 7 Clients.
Index #9: Linked List of 7 Clients.
Index #10: Linked List of 7 Clients.
========================================================================
```

If a new client tried to create an account using the same email, you should show an error message and not add the client to the system. (3 Points)

After creating the account successfully, you will go back to the mainMenu() then the client can login using the created info. (2 Point)

In the clientHome() menu, if the client chose the transactionsMenu() he/she can create a new transaction, in the point you don't have to take info from the user, just create a new object with static data and add the transaction to the array of transactions related to this client, client can't create more than 5 transactions. (3 Points)

To view the transactions, you just need to view the array of transactions, to delete a transaction client must type the transaction id, and if it is correct, delete it. (2 Points)

===

Part #4: Bonus Part (15 Points)

Sort each linked list by the id of the clients, then print the clients of each linked list and the number of clients in that list. (You will create an option in the mainMenu() for this part)

With Gratitude