**SIMPLE BANKING SYSTEM**

This is a simple banking system implemented in C that allows users to create accounts, log in, deposit, withdraw money, check balances, and view transaction history.

**Features:**

* Create an account
* Log in to an account
* Deposit money into an account
* Withdraw money from an account
* Check account balance
* View transaction history

**Structure:**

**Account Structure**

The Account structure contains the following fields:

* **username** - The username of the account holder
* **password** - The password of the account holder
* **balance** - The current balance of the account
* **transaction\_count** - The number of transactions
* **transaction\_capacity** - The capacity of the transactions array
* **transactions** - An array of transactions
* **next** - A pointer to the next account in the linked list

**Functions:**

**create\_account():**

Prompts the user to enter a username and password, and creates a new account with an initial balance of 0. Initializes the transactions array with a capacity of 10.

**authenticate():**

Prompts the user to enter a username and password, and checks if they match any existing account. Returns a pointer to the account if a match is found, otherwise returns NULL.

**deposit(Account\* account):**

Prompts the user to enter an amount to deposit into their account, updates the account balance, and records the transaction.

**withdraw(Account\* account):**

Prompts the user to enter an amount to withdraw from their account, checks if the account has sufficient balance, updates the account balance, and records the transaction.

**check\_balance(Account\* account):**

Prints the current balance of the account.

**transaction\_history(Account\* account):**

Prints the transaction history of the account.

**login():**

Allows the user to log in to their account and perform various operations like depositing money, withdrawing money, checking balance, and viewing transaction history.

**free\_memory():**

Frees all the allocated memory for accounts and transactions before the program exits.