BIL141 Homework Assignment 4

Due date: 27.11.2019 23:59

In this homework assignment, you are to implement a basic banking system which will allow deposit, withdrawal, transfer of funds. To make this implementation, first, you need to implement **Bank** and **Account** structs whose attributes are as follows.

Bank

- **bankName:** name of the bank
- **maxNumberOfAccounts:** maximum number of accounts that will can be created in this bank
- **curNumberOfAccounts:** number of accounts currently in use in this bank
- **allAccounts:** type of **Account**** which will be used to access all accounts in this bank

Account

- **ownerName:** name of the account holder
- **password:** password for the account, will be used when accessing account
- **amount:** amount of money the account has
- **accountCreationYear:** year on which account had been created

Additionally you are expected to implement following functions.

```
Bank* bank_constructor(char* bankName, int maxNumberOfAccounts);
```

- Create a bank with the given arguments and return it.
- Do not forget to allocate memory.

```
void bankInfo(Bank* b);
```

- Shows information about the bank and sum of funds deposited in this bank. Check the example below for formatting.

```
void customerInfo(Bank *b);
```

- Displays information about all customers of this bank. Check the example below for formatting.

```
int createAccount(Bank* b, char* holderName, char* password, int accountCreationYear);
```

- Creates an account in the given bank with given information. Please note that different banks can have the same account number and combination of bank name and account number will be used to access any account. Returns the account number in that bank.
- Don't forget to allocate memory.

```
int depositMoney(Bank* b, int accountNumber, int amount);
```

- Deposits given amount of money to the account with given information. Note that money can be deposited without password. Returns the deposited amount.

```
int withdrawMoney(Bank* b, int accountNumber, char* password, int amount);
```

- If account has sufficient funds and password is correct withdrawal is made. If funds are insufficient or password is wrong print errors according to the examples below. Return the withdrawn amount.

```
int transferMoney(Bank* senderBank, int senderAccount, char* senderPassword, Bank* receiverBank, int receiverAccount, int amount);
```

- If sender account has sufficient funds and password is correct transfer is made. If funds are insufficient or password is wrong print errors according to the examples below. Return the transferred amount.

```
void checkBalance(Bank* b, int accountNumber, char* password);
```

- Display balance or the error message depending the correctness of the password according to the examples below.

Examples

1. This example creates a system with 2 banks and 3 accounts. For debug purposes, use the same setup.

```
    Bank* b1 = bank_constructor("BankA", 3);
    Bank* b2 = bank_constructor("BankB", 4);
    int hesap11 = createAccount(b1, "Ahmet", "pass", 1980);
    int hesap12 = createAccount(b1, "Veli", "word", 2000);
    int hesap21 = createAccount(b2, "Mehmet", "sifre", 1990);
    depositMoney(b1, hesap11, 50);
    depositMoney(b1, hesap12, 80);
    transferMoney(b1, hesap11, "hatali", b2, hesap21, 20); //Transfer with wrong password fails
    transferMoney(b1, hesap11, "pass", b2, hesap21, 20); //Transfer with correct password is successful
    withdrawMoney(b1, hesap12, "hatali", 40); //Withdrawal with wrong password fails
    withdrawMoney(b1, hesap12, "word", 120); //Withdrawal with insufficient funds
    withdrawMoney(b1, hesap12, "word", 40);   //Withdrawal with correct passwords is successful
    checkBalance(b1, hesap11, "hatali"); //Check balance with wrong password fails
    checkBalance(b1, hesap11, "pass"); //Check balance with wrong password is successful
    checkBalance(b2, hesap21, "sifre");
    checkBalance(b1, hesap12, "word");
    customerInfo(b1);
    bankInfo(b2);
```

2. Shows an example for wrong password input.

```
Wrong Password!
```

3. Shows an example for transfer or withdrawal with insufficient funds.

```
Insufficient funds!
```

4. Shows an example of a successful balance check.

```
Account balance: 40
```

5. Shows the output for `customerInfo(b1);` function in the given setup.

```
Customer List   :
Ahmet since 1980.
Veli since 2000.
```

6. Shows the output for `bankInfo(b2)` function in the given setup.

```
Name of the bank: BankB
Total Capital   : 20
```

7. Total output for the given setup is as follows.

```
Wrong Password!
Wrong Password!
Insufficient funds!
Wrong Password!
Account balance: 30
Account balance: 20
Account balance: 40
Customer List   :
Ahmet since 1980.
Veli since 2000.
Name of the bank: BankB
Total Capital   : 20
```