Assignment 10

Due on: November 17^{th}, Sunday 6pm
This Assignment will be Interview Graded

*It would be helpful to finish Lab 8 before starting this assignment.

Task: Automated Teller Machine (ATM) with gambling

For Assignment 10, you will manage a bank database by changing the data depending on users' input values. To make the problem easier, the bank account database has a maximum of 100 accounts, and each account has its owner's name, balance, and password. The ATM also has additional ten accounts for gambling that show only the balance. The initial balance in this account is 0, 500, 10, 1000, 20, 5000, 30, 10000, 40, 50000. Predeclare the account database as follows:

```
string accountName[100] = { };
int accountBalance[100] = {0};
int accountPassword[100] = {0};
int gambling[10] = {0, 500, 10, 1000, 20, 5000, 30, 10000, 40, 50000};
```

For example, if Taeho has a balance of $2000 at index 0 and the password is 1234, accountName[0] = "Taeho", accountBalance[0] = 2000, and accountPassword[0] = 1234. Similarly, if Maciej has a balance of $10000 at index 1 and the password is 5678, accountName[1] = "Maciej", accountBalance[1] = 10000, and accountPassword[1] = 5678.

The program should perform the selected option and then display the menu again so that the user can choose one of the options (Except ***Finish*** button). The menu has the following options:

1. ***Make an account***. This option will allow the user to make an account. The working procedure is as follows:

   (a) If 100 accounts have already been created, display "The bank account database is full" and return to the menu. Otherwise, call a function called `newAccount` which takes in a reference to an accountName element and a reference to an accountPassword element using the currently unused index.

   (b) In the function, ask the user to enter his/her name at first. Then, ask the user to enter the password of the account.

   (c) Lastly, the name and the password are saved in the bank account database. (It is assumed that the same name does not already exist in the database.)

2. ***Deposit***. This option will allow the user to deposit money in his/her account. The working procedure is as follows:

   (a) Ask the user to enter a name.

   (b) Check the database has an account that matches the name. If there is a matching account, call a function called `deposit`, which takes in a reference to an accountBalance element and a value to an accountPassword element using the detected index, and go to the next step. If not, display "No account!" and return to the menu.

   (c) In the `deposit` function, ask the user to enter the password. If the password entered matches the password of the account, go to the next step. If not, display "Wrong password!" and return to the menu.

(d) Ask the user to enter the deposit amount. (Suppose the user enters a normal positive integer amount.) Then, add the deposit amount to the balance of the account.

3. ***Withdraw***. This option will allow the user to withdraw money in his/her account. The working procedure is as follows:

   (a) Ask the user to enter a name.

   (b) Check the database has an account that matches the name. If there is a matching account, call a function called `withdraw`, which takes in a reference to an accountBalance element and a value to an accountPassword element using the detected index, and go to the next step. If not, display "No account!" and return to the menu.

   (c) In the `withdraw` function, ask the user to enter the password. If the password entered matches the password of the account, go to the next step. If not, display "Wrong password!" and return to the menu.

   (d) Show the current balance and ask the user to enter the withdraw amount. (Suppose the user does not enter a negative amount.)

   (e) If the input amount is greater than the current balance, go to the previous step. Otherwise, go to the next step.

   (f) Subtract the withdraw amount from the balance of the account.

4. ***Balance***. This option will show the balance of the user's account. The working procedure is as follows:

   (a) Ask the user to enter a name.

   (b) Check the database has an account that matches the name. If there is a matching account, go to the next step. If not, display "No account!" and return to the menu.

   (c) Ask the user to enter the password. If the password entered matches the password of the account, show the balance of the account. If not, display "Wrong password!".

   (d) Return to the menu.

5. ***Gambling***. This option swaps the balance of randomly selected ATM's gambling account and that of the user's account. The working procedure is as follows:

   (a) Ask the user to enter a name.

   (b) Check the database has an account that matches the name. If there is a matching account, go to the next step. If not, display "No account!" and return to the menu.

   (c) Ask the user to enter the password. If the password entered matches the password of the account, call a function called `gamblingIdx` which takes in nothing and go to the next step. If not, display "Wrong password!" and return to the menu.

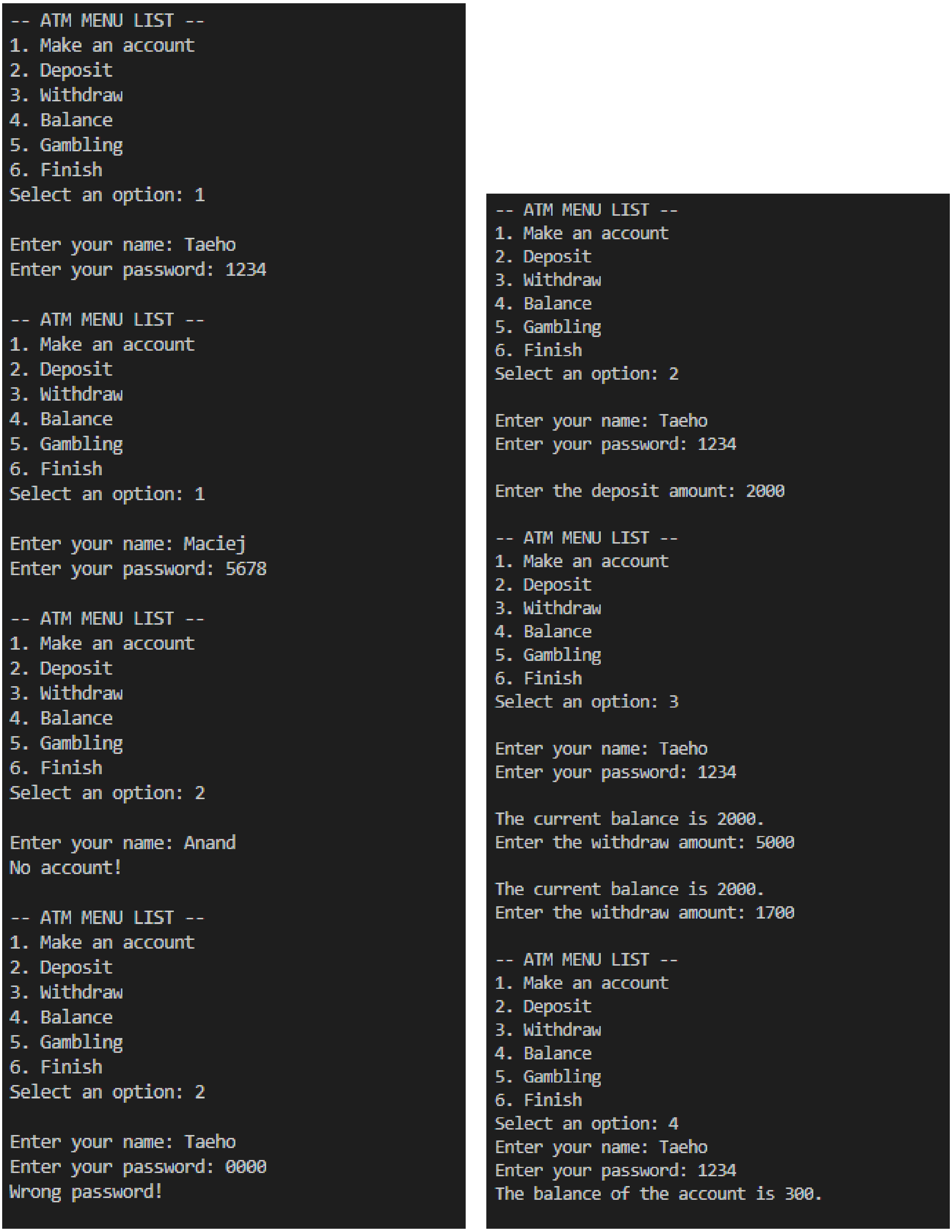
   (d) Show the balance of the account before gambling. If the balance is less than 500, display "The balance is less than the minimum balance for gambling!".

   (e) The `gamblingIdx` function returns the value of one of $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ using the rand() function. The return value obtained from the `gamblingIdx` function becomes the index of the gambling account.

   (f) Call a function `swapValues`, which takes in a reference to an accountBalance element using the user's account index and a reference to an accountPassword element using the gambling index from the `gamblingIdx` function, to swap the balance of the designated gambling account and that of the user's account.

   (g) Show the balance of the account after gambling and return to the menu.

6. ***Finish***. Terminate the program.



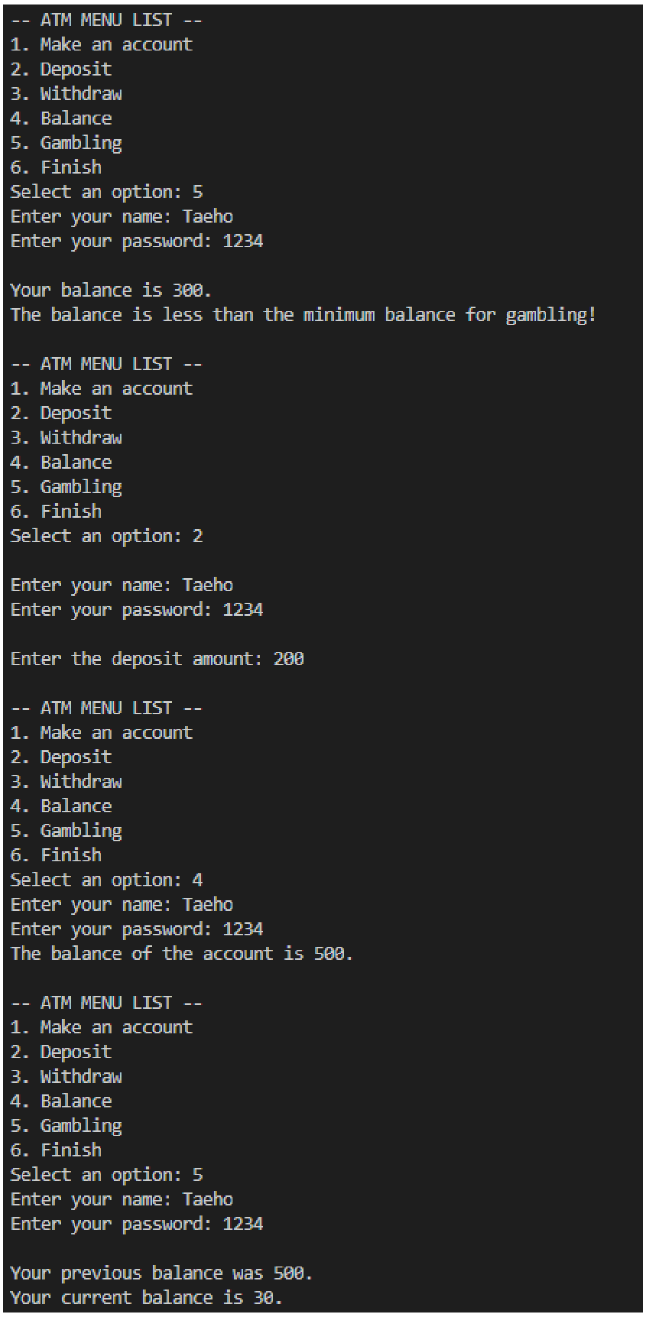
```
-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 1

Enter your name: Taeho
Enter your password: 1234

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 1

Enter your name: Maciej
Enter your password: 5678

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 2

Enter your name: Anand
No account!

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 2

Enter your name: Taeho
Enter your password: 0000
Wrong password!
```

(a) Example (1 of 4)

```
-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 2

Enter your name: Taeho
Enter your password: 1234

Enter the deposit amount: 2000

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 3

Enter your name: Taeho
Enter your password: 1234

The current balance is 2000.
Enter the withdraw amount: 5000

The current balance is 2000.
Enter the withdraw amount: 1700

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 4
Enter your name: Taeho
Enter your password: 1234
The balance of the account is 300.
```

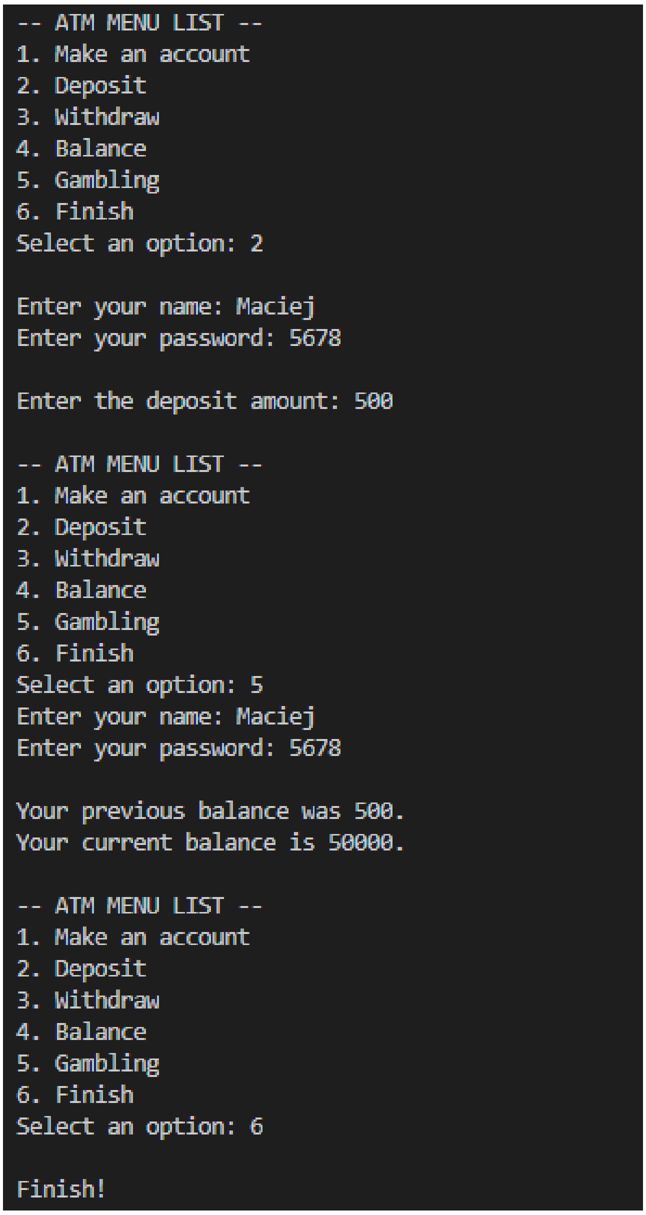
(b) Example (2 of 4)

Figure 1: Example (*Continue*)



```
-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 5
Enter your name: Taeho
Enter your password: 1234

Your balance is 300.
The balance is less than the minimum balance for gambling!

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 2

Enter your name: Taeho
Enter your password: 1234

Enter the deposit amount: 200

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 4
Enter your name: Taeho
Enter your password: 1234
The balance of the account is 500.

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 5
Enter your name: Taeho
Enter your password: 1234

Your previous balance was 500.
Your current balance is 30.
```

(a) Example (3 of 4)



```
-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 2

Enter your name: Maciej
Enter your password: 5678

Enter the deposit amount: 500

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 5
Enter your name: Maciej
Enter your password: 5678

Your previous balance was 500.
Your current balance is 50000.

-- ATM MENU LIST --
1. Make an account
2. Deposit
3. Withdraw
4. Balance
5. Gambling
6. Finish
Select an option: 6

Finish!
```

(b) Example (4 of 4)

Figure 2: Example

Instructions to submit your Assignment

The assignment should contain a fully functional ATM with gambling C++ source code. Submit your .cpp file on Moodle under Assignment 10 by the due date. You do not need to submit any executable files.

Keep in mind the Honor code and ensure that you do not violate any of the rules it entails.