PROGRAMMING ASSIGNMENT #5

This program is similar to programming assignment #4. That is, you will have all of the requirements as in Programming Assignment #4, including data error checking on all inputs.

Your program is to behave in the exact same manner as assignment #4, however how you structure your code is a different story.

Now we will use ***functions***.

Recall Programming Assignment #4:

Write a C program that allows the user to make some simple ***banking transactions***. The program should first prompt the user to enter the **current balance** of his/her bank account (in dollars and cents, *not less than zero*). The program should then prompt the user to enter the **number of withdrawals** to make, and then the **number of deposits** to make. For this assignment, let's set a ***maximum of 5 deposits and 5 withdrawals*** *(etc.)*

Here is the change for program #5:

There should be at least two non-trivial functions in the program. (Non-trivial means that the function does more than simply print a line of text.)

Here are some examples of what you might want to use for your functions:

Input functions:

1. Obtain opening (current) balance.
2. Obtain number the number of deposits.
3. Obtain number of withdrawals.
4. Obtain deposit amounts.
5. Obtain withdrawal amounts.

Output functions:

1. Display closing balance and associated message.
2. Display message based on closing balance.
3. Display bank record.

Hints: Your code from assignment #4 does not need to be modified *too* much to solve this problem. The algorithm is the same. Outputting the contents of the arrays in "record form" is pretty straightforward, and should be done in a loop. (This *could* be a good place for a function.)

The most complicated of the above functions is the one which would **display the bank record**. If you decide that you want to attempt this one, I will give you a head start by providing the function header as shown here:

void (display_bank_record (float start_balance, float deposits [ ], int num_deposits, float withdrawals [ ], int num_withdrawals, float end_balance)

Good luck!