```
**************************************************************
*   PROGRAMMED BY : Andrew Gharios
*   STUDENT ID    : 1449366
*   CLASS         : M-Th 5-7:20p
*   ASSIGNMENT #1 : Functions and Arrays
*****************************************************************
What input file would you like to use?   InFile.txt
What output file would you like to use?  OFile.txt
MENU OPTIONS

1 - Find the larger balance
2 - Find the smaller balance
3 - Obtain the sum of all balances
4 - Obtain the average of all balances
5 - Find person
0 - Exit
Enter an option (0 to exit): 1

Finding the larger balance...

<redisplay menu>
Enter an option (0 to exit): 2

Finding the smaller balance...

<redisplay menu>
Enter an option (0 to exit): 3

Obtaining the sum of all balances...

<redisplay menu>
Enter an option (0 to exit): 4

Obtaining the average of all balances...

<redisplay menu>
Enter an option (0 to exit): 5

What do you want to search for (enter done to exit): Steve Woolston
Found.

<redisplay menu>
Enter an option (0 to exit): 5

What do you want to search for (enter done to exit): Jacques Rousseau
Jacques Rousseau was not found.

<redisplay menu>
Enter an option (0 to exit): 5
```

```

What do you want to search for (enter done to exit): Chris Carroll
Found.

<redisplay menu>
Enter an option (0 to exit): 5

What do you want to search for (enter done to exit): Pete McBride
Found.

<redisplay menu>
Enter an option (0 to exit): 5

What do you want to search for (enter done to exit): Jean Rousseau
Found.

<redisplay menu>
Enter an option (0 to exit): 5

What do you want to search for (enter done to exit): Florence Cyr
Florence Cyr was not found.

<redisplay menu>
Enter an option (0 to exit): 0

Thank you for using my program.
```

```
***************************************************************
*   PROGRAMMED BY : Andrew Gharios
*   STUDENT ID    : 1449366
*   CLASS         : M-Th 5-7:20p
*   ASSIGNMENT #1 : Functions and Arrays
******************************************************************
Larger Balance:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1002     Steve Woolston           $   1423.20

Smaller Balance:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1003     Don McBride              $     12.32

Sum of Balance for all persons:
$  4080.48

Average of Balance for all persons:
$   408.05

Search Name:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1002     Steve Woolston           $   1423.20

Search Name:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1008     Chris Carroll            $     32.35

Search Name:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1007     Pete McBride             $    500.32

Search Name:
ID #     NAME                     BALANCE DUE
----     --------------------     -----------
1001     Jean Rousseau            $     15.50


```

```
/*********************************************************************************
* AUTHOR     : Andrew Gharios
* STUDENT ID : 1449366
* AS #1      : Functions and Arrays
* CLASS      : CS1B
* SECTION    : M-TH: 5-7:20p
* DUE DATE   : 6/15/21
*********************************************************************************/
#ifndef HEADER_H_
#define HEADER_H_

#include <iostream> // cin, cout.
#include <string> // string datatype variables.
#include <fstream> // Fstream files.
#include <iomanip> // fixed, setw, setprecision.
#include <ostream> // Ostream data type.
using namespace std;

/********************************************************************************
 * File To Array
 *  This function will take all the data from the Input file and place them
 *  into their respective Array with the corresponding datatype.
 *  ==> returns nothing
 *
********************************************************************************/
void FileToArray(ifstream& InFile,
                 const int AR_SIZE,
                 string    ArrayName[], // IN & OUT - Array of the names.
                 int   ArrayID[],   // IN & OUT - Array of the Ids.
                 float     ArrayBal[]); // IN & OUT - Array of the balances.

/********************************************************************************
 * Search Balance
 *  This function will search for the smaller or the larger balance in the
 *  balances Array based on user selection. Then output the person and all their
 *  information with the smaller or larger balance.
 *  ==> returns nothing
 *
********************************************************************************/
void SearchBalance(ofstream& OFile,
                   const int AR_SIZE,
                   const float     ArrayBal[],
                   const int       ArrayIds[],
                   const string    ArrayName[], // IN & OUT - Array of the
                   balances.
                   bool      larger);

/
  ***************************************************************************
  ****
 * Search Name
 *  This function will search for a name within the names Array and then
   output
 *  the all the information of that found name to an output file.
 *  ==> returns nothing
 *

*****************************************************************************
**/
void SearchName(ofstream& OFile,
                const int AR_SIZE,
                string  ArrayName[],
                int  ArrayID[],
                float ArrayBal[]);   // IN & OUT - Array of the balances.

/
  ***************************************************************************
  ****
 * Sum Avg.
 *  This function will calculate the sum or the average of all the balances
   in
 *  passed array.
 *  ==> returns nothing
 *

*****************************************************************************
**/
void SumAvg(ofstream& OFile,
            const int AR_SIZE,
            float     ArrayBal[], // IN & OUT - Array of the balances.
            bool      average);

/
  ***************************************************************************
  ****
 * PrintHeaderFile
 *   This function will output the header information
 *

*****************************************************************************
**/
void PrintHeaderFile(ostream& output,      // IN - output datatype.
                     ofstream& OFile,      // IN - Output to File variable.
                     string asName,        // IN - assignment name
                     int asNum,            // IN - assignment number
                     string studentName,   // IN - student's name
                     string classInfo,     // IN - class that is being taken
                     char asType,          // IN - assignment type
                     long long studentID); // IN - student ID

#endif

```

```
/*******************************************************************************
* AUTHOR     : Andrew Gharios
* STUDENT ID : 1449366
* AS #1      : Functions and Arrays
* CLASS      : CS1B
* SECTION    : M-TH: 5-7:20p
* DUE DATE   : 6/15/21
*******************************************************************************/
#include "Header.h"

/*******************************************************************************
* Functions and Arrays
*------------------------------------------------------------------------------
* This program will take in which files user wants to use and then prompt a
* series of option for the user to be able to use the data in the file for
* specific tasks. The options include searching for a name within the list,
* comparing the balances and determining the largest or smallest, calculating
* the total and average of all balances. The program outputs a list at to the
* OFile with all request information.
*------------------------------------------------------------------------------
* INPUT:
* UserIfFile - User's selected InFile.
* UserOfFile - User's selected OFile.
* option     - User's selection of options.
*******************************************************************************/
int main()
{
    /***************************************************************************
    * CONSTANTS
    * -------------------------------------------------------------------------
    * OUTPUT - USED FOR CLASS HEADING
    * -------------------------------------------------------------------------
    * PROGRAMMER : Programmer's Name
    * CLASS    : Student's Course
    * SECTION     : Class Days and Times
    * LAB_NUM    : Lab Number (specific to this lab)
    * LAB_NAME   : Title of the Lab
    * -----------------------------------------------------------------------
    * INPUT_COL : Setw size for prompt column.
    * AR_SIZE   : Array standard size.
    **************************************************************************/

    const string AS_NAME = "Functions and Arrays";
    const int AS_NUM = 1;
    const string STUDENT_NAME = "Andrew Gharios";
    const string CLASS_INFO = "M-Th 5-7:20p";
    const char AS_TYPE = 'A';
    const long long STUDENT_ID = 1449366;

    const int AR_SIZE   = 10;
    const int INPUT_COL = 41;

    string ArNames[AR_SIZE]; // IN & CALC - Array of the names.
    int    ArIds[AR_SIZE];   // IN & CALC - Array of the ids.
    float  ArBal[AR_SIZE];   // IN & CALC - Array of the balances.
    ifstream InFile;     // IN & CALC - Input File variable.
    ofstream OFile;          // IN & CALC - Output File variable.
    string   userIfile;      // IN & CALC - Input File selection from user.
    string   userOfile;      // IN & CALC - Output File selection from user.
    int      option;         // IN & CALC - User selection of what to do with data.
    bool     larger;         // IN & CALC - If user wants to find the larger or smaller balance.
    bool     average;        // IN & CALC - If user wwants to calculate sum or average.

    option = 999;

    PrintHeaderFile(cout, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
                    AS_TYPE, STUDENT_ID);

    cout << left;
    cout << setw(INPUT_COL) << "What input file would you like to use?";
    getline(cin, userIfile);
    cout << setw(INPUT_COL) << "What output file would you like to use?";
    getline(cin, userOfile);

    InFile.open(userIfile);
    OFile.open(userOfile);

    PrintHeaderFile(OFile, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
        AS_TYPE, STUDENT_ID);

    FileToArray(InFile, AR_SIZE, ArNames, ArIds, ArBal);


    cout << "MENU OPTIONS" << endl << endl;
    cout << "1 - Find the larger balance" << endl;
    cout << "2 - Find the smaller balance" << endl;
    cout << "3 - Obtain the sum of all balances" << endl;
    cout << "4 - Obtain the average of all balances" << endl;
    cout << "5 - Find person" << endl;
    cout << "0 - Exit" << endl;

    while (option != 0)
    {
        cout << "Enter an option (0 to exit): ";
        cin  >> option;
        cin.ignore(10000, '\n');

        if (option == 1)
        {
            cout << endl << "Finding the larger balance...";
            cout << endl << endl;
            larger = true;
            SearchBalance(OFile, AR_SIZE, ArBal, ArIds, ArNames, larger);
        }
        else if (option == 2)
        {
            cout << endl << "Finding the smaller balance...";
            cout << endl << endl;
            larger = false;
            SearchBalance(OFile, AR_SIZE, ArBal, ArIds, ArNames, larger);
        }
        else if (option == 3)
        {
            cout << endl << "Obtaining the sum of all balances...";
            cout << endl << endl;
            average = false;
            SumAvg(OFile, AR_SIZE, ArBal, average);
        }
        else if (option == 4)
        {
            cout << endl << "Obtaining the average of all balances...";
            cout << endl << endl;
            average = true;
            SumAvg(OFile, AR_SIZE, ArBal, average);
        }
        else if (option == 5)
        {
            cout << endl;
            SearchName(OFile, AR_SIZE, ArNames, ArIds, ArBal);
        }

        if (option != 0)
        {
            cout << "<redisplay menu>" << endl;
        }
    }

    InFile.close();
    OFile.close();

    cout << endl << "Thank you for using my program.";

    return 0;

}
```

```cpp
#include "Header.h"

/******************************************************************************
 * File To Array
 *  This function will take all the data from the Input file and place them
 *  into their respective Array with the corresponding datatype.
 *
 * INPUTS:
 *  InFile    - Input File.
 *  AR_SIZE   - Array size
 *  ArrayName - Array for all the names.
 *  ArrayID   - Array for all the Ids.
 *  ArrayBal  - Array for all the balances.
 *
 * No Outputs.
 ******************************************************************************/
void FileToArray(ifstream& InFile,      // IN - Input File variable.
                 const int AR_SIZE,     // IN - Array sizes.
                 string    ArrayName[], // IN - Array for Names.
                 int   ArrayID[],   // IN - Array for Ids.
                 float     ArrayBal[])  // IN - Array for Balances.
{
    int    index; // CALC - Index to manipulate Arrays.
    index = 0;

    while (InFile && index < AR_SIZE)
    {
        getline(InFile, ArrayName[index]);
        InFile >> ArrayID[index];
        InFile >> ArrayBal[index];
        InFile.ignore(10000, '\n');
        index++;
    }
}
```

```cpp
#include "Header.h"

/*******************************************************************************
 * Search Balance
 *  This function will search for the smaller or the larger balance in the
 *  balances Array based on user selection. Then output the person and all their
 *  information with the smaller or larger balance.
*
* INPUTS:
*  OFile      - Output File.
*  AR_SIZE    - Array size
*  ArrayBal   - Array for all the balances.
*  ArrayName - Array for all the names.
*  ArrayID    - Array for all the Ids.
*  average    - Bool to calculate avg or sum.
*
* No Outputs.
*******************************************************************************/
void SearchBalance(ofstream& OFile,             // IN - Output File         variable.
                   const int AR_SIZE,           // IN - Array sizes.
                   const float     ArrayBal[],  // IN - Array for Balances.
                   const int       ArrayIds[],  // IN - Array for Ids.
                   const string    ArrayName[], // IN - Array for Names.
                   bool      larger)        // IN - bool to calculate larger or smaller bal.
{
    const int ID_COL = 9;     // CALC - Setw size for ID column.
    const int NAME_COL = 25;  // CALC - Setw size for Name column.
    const int BAL_COL = 10;   // CALC - Setw size for Balance column.

    int    index;      // CALC - Index to manipulate Arrays.
    float  largerBal;  // CALC - Largest balance storage.
    float  smallerBal; // CALC - Smallest balance storage.
    int    location;   // CALC - Location to store index for output.

    index      = 0;
    largerBal  = 0;
    location   = 0;
    smallerBal = 10000;


    while (index < AR_SIZE)
    {
        if (larger)
        {
            if (largerBal < ArrayBal[index])
            {
                largerBal = ArrayBal[index];
                location  = index;
            }
        }
        else
        {

            if (smallerBal > ArrayBal[index])
            {
                smallerBal = ArrayBal[index];
                location   = index;
            }
        }
        index++;
    }

    OFile << left;
    OFile << fixed;
    OFile << setprecision(2);

    if (larger)
    {
        OFile << "Larger Balance:" << endl;
        OFile << setw(ID_COL)      << "ID #" << setw(NAME_COL);
        OFile << "NAME"            << "BALANCE DUE"  << endl;

        OFile << setw(ID_COL)      << "----";
        OFile << setw(NAME_COL)    << "--------------------";
        OFile << "-----------"     << endl;

        OFile << setw(ID_COL)   << ArrayIds[location];
        OFile << setw(NAME_COL) << ArrayName[location];
        OFile << "$"   << right << setw(BAL_COL) << ArrayBal[location];
        OFile << endl  << endl;
        OFile << setprecision(6);
    }
    else
    {
        OFile << "Smaller Balance:" << endl;
        OFile << setw(ID_COL)       << "ID #" << setw(NAME_COL);
        OFile << "NAME"             << "BALANCE DUE" << endl;

        OFile << setw(ID_COL) << "----";
        OFile << setw(NAME_COL) << "--------------------";
        OFile << "-----------" << endl;

        OFile << setw(ID_COL)   << ArrayIds[location];
        OFile << setw(NAME_COL) << ArrayName[location];
        OFile << "$"  << right  << setw(BAL_COL) << ArrayBal[location];
        OFile << endl << endl;
        OFile << setprecision(6);
    }
}

```

```cpp
#include "Header.h"

/*******************************************************************************
* Search Name
*   This function will search for a name within the names Array and then output
*   the all the information of that found name to an output file.
*
* INPUTS:
*  OFile - Output File.
*  AR_SIZE - Array size
*  ArrayBal - Array for all the balances.
*  ArrayName - Array for all the names.
*  ArrayID - Array for all the Ids.
*  average - Bool to calculate avg or sum.
*
*No Outputs.
*******************************************************************************/
void SearchName(ofstream& OFile, // IN - Output File variable.
                const int AR_SIZE,   // IN - Array sizes.
                string  ArrayName[], // IN - Array for Names.
                int  ArrayID[],      // IN - Array for Ids.
                float ArrayBal[])    // IN - Array for Balances.
{
    const int ID_COL = 9;     // CALC - Setw size for ID column.
    const int NAME_COL = 25;  // CALC - Setw size for Name column.
    const int BAL_COL = 10;   // CALC - Setw size for Balance column.

    int     index;    // CALC      - Index to manipulate Arrays.
    string  userPick; // IN & CALC - User's pick for which name to search.
    boolfound;    // CALC      - If name was found or not.

    found = false;
    index = 0;

    cout << "What do you want to search for (enter done to exit): ";
    getline(cin, userPick);

    while (index < AR_SIZE && !found)
    {
        if (userPick == ArrayName[index])
        {
            found = true;
            cout << "Found." << endl << endl;

            OFile << left;
            OFile << fixed;
            OFile << setprecision(2);

            OFile << "Search Name:" << endl;
            OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
            OFile << "NAME"       << "BALANCE DUE" << endl;

            OFile << setw(ID_COL) << "----";
            OFile << setw(NAME_COL) << "---------------------";
            OFile << "-----------" << endl;

            OFile << setw(ID_COL)   << ArrayID[index];
            OFile << setw(NAME_COL) << ArrayName[index];
            OFile << "$"  << right  << setw(BAL_COL) << ArrayBal[index];
            OFile << endl << endl;
        }
        else
        {
            index++;
        }
    }
    if (!found)
    {
        cout << userPick << " was not found." << endl << endl;
    }
}

```

```cpp
#include "Header.h"


/*******************************************************************************
* Sum Avg.
*   This function will calculate the sum or the average of all the balances in
*   passed array.
*
* INPUTS:
*  OFile     - Output File.
*  AR_SIZE   - Array size
*  ArrayBal  - Array for all the balances.
*  average   - Bool to calculate avg or sum.
*
* No Outputs.
*******************************************************************************/
void SumAvg(ofstream& OFile,      // IN - Output File variable.
            const int AR_SIZE,    // IN - Array sizes.
            float     ArrayBal[], // IN - Array for Balances.
            bool      average)    // IN - Bool to calculate avg or sum.
{
    int   index;    // CALC        - Index to manipulate Arrays.
    float totalBal; // CALC & OUT - Sum of all balances.
    float avg;      // CALC & OUT - Avg of all balances.

    totalBal = 0;

    for (index = 0; index < AR_SIZE; index++)
    {
        totalBal += ArrayBal[index];
    }

    OFile << setprecision(2);
    OFile << fixed;

    if (average)
    {
        avg = totalBal / AR_SIZE;
        OFile << "Average of Balance for all persons:" << endl;
        OFile << "$" << setw(9) << avg;
        OFile << endl << endl;
    }
    else
    {

        OFile << "Sum of Balance for all persons:" << endl;
        OFile << "$"  << setw(9) << totalBal;
        OFile << endl << endl;
    }
    OFile << setprecision(6);
}
```

```cpp
#include "Header.h"

/*******************************************************************************
 * PrintHeaderFile
 *   This function will output the header information
 *_______________________________________________________________________________
 * PRE-CONDITIONS
 *   The following parameters need to have a defined value prior to calling
 *   the function
 *          asName: The name of the assignment given in the course
 *          asNum: The number of the assignment given in the course
 *          studentName: The name of the student writing the code
 *          classInfo: The course name, date, and time of the class
 *          asType: Will either output as a lab or an assignment
 *          studentID: The Identification Number of the student
 *******************************************************************************/

void PrintHeaderFile(ostream& output,     // IN - output datatype.
                     ofstream& OFile,     // IN - Output to File variable.
                     string asName,       // IN - assignment name
                     int asNum,           // IN - assignment number
                     string studentName,  // IN - student's name
                     string classInfo,    // IN - class that is being taken
                     char asType,         // IN - assignment type
                     long long studentID) // IN - student ID
{
    output << left;
    output << "***************************************************************\n";
    output << "*   PROGRAMMED BY : " << studentName << endl;
    output << "*   " << setw(14) << "STUDENT ID "   << ": " << studentID << endl;
    output << "*   " << setw(14) << "CLASS "        << ": " << classInfo << endl;
    output << "*   ";

    // PROCESSING - This will adjust setws and format appropriately based
    //              on if this is a lab 'L' or assignment

    if (toupper(asType) == 'L')
    {
        output << "LAB #" << setw(9);
    }
    else
    {
        output << "ASSIGNMENT #" << setw(2);
    }
    output << asNum << ": " << asName << endl;
    output << 
      "*****************************************************************";
    output << right << endl;

    return;
}
```

```
Jean Rousseau
1001 15.50
Steve Woolston
1002 1423.20
Michele Rousseau
1005 52.75
Pete McBride
1007 500.32
Florence Rousseau
1010 1323.33
Lisa Covi
1009 332.35
Don McBride
1003 12.32
Chris Carroll
1008 32.35
Yolanda Agredano
1004 356.00
Sally Sleeper
1006 32.36
```