```
************************************************
* PROGRAMMED BY : Faris Hijazi
* CLASS         : CS1A
* SECTION       : MW: 7:30P
* Assignment #1 : Functions and Arrays
************************************************

What input file would you like to use:  input.txt
What output file would you like to use: output.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…

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): 2

Finding the smaller balance…

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): 3

Finding the larger sum…

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): 4

Finding the larger average…
```

```

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): 5

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

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): 5

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

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): 5

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

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): 5

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

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): 6

```

```
************************************************
* PROGRAMMED BY : Faris Hijazi
* CLASS         : CS1A
* SECTION       : MW: 7:30P
* Assignment #1 : Functions and Arrays
************************************************

Larger Balance:
ID #     NAME                         BALLANCE DUE
----     --------------------         -----------
1002     Steve Woolston               $   1423.20

Smaller Balance:
ID #     NAME                         BALLANCE DUE
----     --------------------         -----------
1002     Steve Woolston               $   1423.20

Sum of Balance for all persons:
$   4080.48

Average of Balance for all persons:
$    408.05

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

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

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

header.h

```
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/
#include <iostream>
#include <iomanip>
#include <cstring>
#include <sstream>
#include <string>
#include <fstream>
using namespace std;

//ENUMERATED TYPES
enum menuoption {EXIT,
                 LARGERBALANCE,
                 SMALLERBALANCE,
                 SUM,
                 AVERAGE,
                 SEARCH};

//PROTOTYPES
void printHeader(ostream &output,     //output device
                 char exersize,       //lab or assignment?
                 string exersizeName, //lab or assignment name
                 int num,             //lab or assignment name
                 string names         //names of programmer(s)
                 );

void input(string &outFileName,       //name of output file
           string ar1[],              //string array
           int ar2[],                 //int array
           float ar3[],               //float array
           const int AR_SIZE          //size of parallel arrays
           );

void menu();

float sumAvg(menuoption token,        //menu option chosen
             float ar3[],             //float array
             const int AR_SIZE        //size of array
             );

int searchBalance(menuoption token,   //menu option chosen
                  float ar3[],        //float array
                  const int AR_SIZE   //size of array
                  );

int search(menuoption token,          //menu option chosen
           string ar1[],              //string array
           const int AR_SIZE,         //size of array
           string userInput,          //user inputed search item
           bool &found                //returns true if found or false if not found
           );
```

```cpp
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/
/******************************************************************************
 * Functions and Arrays
 * ----------------------------------------------------------------------------
 * This program will receive an input file with the names, ids, and balances
 * of users, the program will prompt the user with a menu which will let the user
 * see the largest/smallest balance sum of all balances average of all balances
 * or search the array for a specific person. The results of this are output to
 * the specified output file.
 * ----------------------------------------------------------------------------
 * INPUT:
 *          input file and menu option
 * OUTPUT:
 *          output file
 *****************************************************************************/
#include "header.h"

int main()
{
    const int AR_SIZE = 10;

    ofstream oFile;        //OUT         - output file variable
    string outfile;        //IN          - name of output file
    string name[10];       //OUT         - name array
    int id[10];            //OUT         - id array
    float balance[10];     //OUT         - balances array
    int menuInput;         //IN & CALC   - menu item chosen
    int searchIndex;       //OUT         - index of search item
    bool nameFound;        //CALC        - name found/not found
    string nameSearch;     //IN          - name to search for
    menuoption menuChoice; //IN          - menu item chosen

    printHeader(cout, 'A', "Functions and Arrays", 1, "Faris Hijazi");

    input(outfile, name, id, balance, AR_SIZE);

    oFile.open(outfile);

    printHeader(oFile, 'A', "Functions and Arrays", 1, "Faris Hijazi");

    menu();
    cin >> menuInput;
    cin.ignore(1000, '\n');
    menuChoice = menuoption(menuInput);

    while(menuChoice > 0 && menuChoice < 6)
    {
        switch(menuChoice)
        {
            case EXIT           : break;

            case LARGERBALANCE  : searchIndex = searchBalance(LARGERBALANCE, balance, AR_SIZE);
                                  cout  << "\nFinding the larger balance…\n\n";
                                  oFile << "Larger Balance:\n";
                                  oFile << left << setw(9) << "ID #"
                                        << setw(25) << "NAME" << "BALLANCE DUE\n";
                                  oFile << left << setw(9) << "----"
                                        << setw(25) << "--------------------" << "-----------\n";
                                  oFile << left << setw(9) << id[searchIndex] << setw(25)
                                        << name[searchIndex];
                                  oFile <<  "$" << setprecision(2) << right << fixed << setw(10) << balance[searchIndex]
                                        << endl << endl << setprecision(6);
                                  oFile.unsetf(ios::fixed);
                                  break;

            case SMALLERBALANCE : searchIndex = searchBalance(LARGERBALANCE, balance, AR_SIZE);
                                  cout  << "\nFinding the smaller balance…\n";
                                  oFile << "Smaller Balance:\n";
                                  oFile << left << setw(9) << "ID #"
                                        << setw(25) << "NAME" << "BALLANCE DUE\n";
                                  oFile << left << setw(9) << "----"
                                        << setw(25) << "--------------------" << "-----------\n";
                                  oFile << left << setw(9) << id[searchIndex] << setw(25)
                                        << name[searchIndex];
                                  oFile <<  "$" << setprecision(2) << right << fixed << setw(10) << balance[searchIndex]
                                        << endl << endl << setprecision(6);
                                  oFile.unsetf(ios::fixed);
                                  break;

            case SUM            : cout  << "\nFinding the larger sum…\n";
                                  oFile << "Sum of Balance for all persons:\n";
                                  oFile << setprecision(2) << fixed;
                                  oFile << "$" << setw(10) << sumAvg(SUM, balance, AR_SIZE);
                                  oFile << endl << endl;
                                  oFile << setprecision(6);
                                  oFile.unsetf(ios::fixed);
                                  break;

            case AVERAGE        : cout  << "\nFinding the larger average…\n";
                                  oFile << "Average of Balance for all persons:\n";
                                  oFile << setprecision(2) << fixed;
                                  oFile << "$" << setw(10) << sumAvg(AVERAGE, balance, AR_SIZE);
                                  oFile << endl << endl;
                                  oFile << setprecision(6);
                                  oFile.unsetf(ios::fixed);
                                  break;

            case SEARCH         : cout << "\nWho do you want to search for (enter done to exit): ";
                                  getline(cin, nameSearch);

                                  searchIndex = search(LARGERBALANCE, name, AR_SIZE, nameSearch, nameFound);

                                  if(nameFound)
                                  {
                                      oFile << "Search Name:\n";
                                      oFile << left << setw(9) << "ID #"
                                            <<setw(25) << "NAME" << "BALLANCE DUE\n";
                                      oFile << left << setw(9) << "----"
                                            <<setw(25) << "---------------------" << "-----------\n";
                                      oFile << left << setw(9) << id[searchIndex] << setw(25)
                                            << name[searchIndex];
                                      oFile <<  "$" << setprecision(2) << right << fixed << setw(10) << balance[searchIndex]
                                            << endl << endl << setprecision(6);
                                      oFile.unsetf(ios::fixed);
                                   }
                                  break;
        }
        menu();
        cin >> menuInput;
        cin.ignore(1000, '\n');
        menuChoice = menuoption(menuInput);
    }
    return(0);
}

```

searchBalance.cpp

```cpp
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/

/******************************************************************************
 * This function will search a given array for the largest or smallest floating
 * point value and return the index it is located at.
 *----------------------------------------------------------------------------
 * INPUT:
 *          token   - menu option chosen
 *          ar3[]   - float array
 *          AR_SIZE - size of array
 * OUTPUT:
 *          i       - index of largest or smallest value
 *****************************************************************************/
#include "header.h"

int searchBalance(menuoption token, float ar3[], const int AR_SIZE)
{

    int index;
    int i;
    float target;

    target = ar3[0];
    i = 0;

    if(token == LARGERBALANCE)
    {
        for(index=0; index < AR_SIZE - 1; index++)
        {
            if(ar3[index + 1] > target)
            {
                target = ar3[index + 1];
                i = index +1;
            }
        }
    }
    else
    {
        for(index=0; index < AR_SIZE - 1; index++)
        {
            if(ar3[index + 1] < target)
            {
                target = ar3[index + 1];
                i = index +1;
            }
        }
    }
    return(i);
}

```

search.cpp

```cpp
/*****************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 ****************************************************************************/

/*****************************************************************************
 * This function will search the an array of type string for a specified string
 * and return the index of the match if found
 *---------------------------------------------------------------------------
 * INPUT:
 *          token     - menu option chosen
 *          ar1[]     - string array
 *          AR_SIZE   - size of array
 *          userInput - user inputed search item
 * OUTPUT:
 *          index     - index returned is that of the matched string
 *          found     - true if match false otherwise
 ****************************************************************************/
#include "header.h"

int search(menuoption token, string ar1[], const int AR_SIZE, string userInput, bool &found)
{
    int index; //CALC & OUT - used in while loop and returned

    index = 0;
    found = false;

    while(!found && index < AR_SIZE)
    {
        if(ar1[index] == userInput)
        {
            found = true;
            cout << "Found\n";
        }
        else
        {
            index++;
        }
    }

    if(!found)
    {
        cout << userInput << "was not found.\n";
    }

    return(index);
}
```

```
/**************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *************************************************************************/

/**************************************************************************
 * This function will find the sum or average of all nums in an array of type
 * float
 *-------------------------------------------------------------------------
 * INPUT:
 *          token   - menu option chosen
 *          ar3[]   - float array
 *          AR_SIZE - size of array
 * OUTPUT:
 *          avg or sum
 *
 *************************************************************************/
#include "header.h"

float sumAvg(menuoption token, float ar3[], const int AR_SIZE)
{
    int index; //CALC - used in for loop
    float sum; //CALC - sum of all values in ar3[]
    float avg; //CALC - avg of all values in ar3[]

    sum = 0;

    for(index = 0; index < AR_SIZE; index++)
    {
        sum += ar3[index];
    }

    if(token == AVERAGE)
    {
        avg = sum/AR_SIZE;
        return(avg);
    }
    else
    {
        return(sum);
    }
}
```

```
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/

/******************************************************************************
 * This function will propagate three arrays of type string, int, and float
 * with values from an input file.
 *----------------------------------------------------------------------------
 * INPUT:
 *          ar1[]       - string array
 *          ar2[]       - int array
 *          ar3[]       - float array
 *          inFile      - input file
 * OUTPUT:
 *          ar1[]       - string array
 *          ar2[]       - int array
 *          ar3[]       - float array
 *          outFileName - name of output file
 *****************************************************************************/

#include "header.h"

void input(string &outFileName, string ar1[], int ar2[], float ar3[], const int AR_SIZE)
{
    ifstream inFile;   //IN    - input file var
    string inFileName; //IN    - name of input file
    int index;         //CALC  - lcv for while loop

    cout << left << setw(40) << "What input file would you like to use:";
    getline(cin, inFileName);

    cout << left << setw(40) << "What output file would you like to use:";
    cout << right;
    getline(cin, outFileName);

    inFile.open(inFileName);

    index = 0;

    while(index < AR_SIZE)
    {
        getline(inFile, ar1[index], '\n');
        inFile >> ar2[index];
        inFile >> ar3[index];
        inFile.ignore(1000, '\n');
        index++;
    }

    inFile.close();
}


```

printHeader.cpp

```cpp
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/

#include "header.h"

/******************************************************************************
 * This function will output the class header using ostream
 *----------------------------------------------------------------------------
 * INPUT:
 *      output      - output file variable
 *      exersize    - Lab or Assignment
 *      exersizeName- name of exersize
 *      num         - number of Lab/Assignment
 *      names       - names of programmers
 * OUTPUT:
 *      header
 *****************************************************************************/
void printHeader(ostream &output, char exersize, string exersizeName,  int num, string names)
{

    int colWidth;  //CALC - changes based on exersize
    string asType; //CALC - changes based on exersize

    if(exersize == 'L')
    {
        asType = "Lab";
        colWidth = 9;
    }
    else
    {
        asType = "Assignment";
        colWidth = 2;
    }

    output << left;
    output <<"************************************************\n";
    output <<"* PROGRAMMED BY : "  << names << endl;
    output <<"* "<< setw(14) << "CLASS"   << ": " << "CS1A"    << endl;
    output <<"* "<< setw(14) << "SECTION" << ": " << "MW: 7:30P"  << endl;
    output <<"* "<< asType << " #" << setw(colWidth)  << num   << ": " << exersizeName << endl;
    output <<"************************************************\n\n";
    output << right;
}

```

menu.cpp

```cpp
/******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * ASSIGNMENT 1 : Functions and Arrays
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 02/15/19
 *****************************************************************************/

/******************************************************************************
 * This function will output the menu
 *----------------------------------------------------------------------------
 * INPUT:
 *          NA
 * OUTPUT:
 *          menu
 *****************************************************************************/
#include "header.h"

void menu()
{
    cout << "\nMENU OPTIONS\n\n";
    cout << "1 - Find the larger balance\n";
    cout << "2 - Find the smaller balance\n";
    cout << "3 - Obtain the sum of all balances\n";
    cout << "4 - Obtain the average of all balances\n";
    cout << "5 - Find person\n";
    cout << "0 - Exit\n";
    cout << "Enter an option (0 to exit): ";
}

```