```
 1 ********************************************
 2 * PROGRAMMED BY : Faris Hijazi
 3 * CLASS         : CS1A
 4 * SECTION       : MW: 7:30P
 5 * Assignment #1 : Functions and Arrays
 6 ********************************************
 7
 8 What input file would you like to use:  input.txt
 9 What output file would you like to use: output.txt
10
11 MENU OPTIONS
12
13 1 - Find the larger balance
14 2 - Find the smaller balance
15 3 - Obtain the sum of all balances
16 4 - Obtain the average of all balances
17 5 - Find person
18 0 - Exit
19 Enter an option (0 to exit): 1
20
21 Finding the larger balance…
22
23 MENU OPTIONS
24
25 1 - Find the larger balance
26 2 - Find the smaller balance
27 3 - Obtain the sum of all balances
28 4 - Obtain the average of all balances
29 5 - Find person
30 0 - Exit
31 Enter an option (0 to exit): 2
32
33 Finding the smaller balance…
34
35 MENU OPTIONS
36
37 1 - Find the larger balance
38 2 - Find the smaller balance
39 3 - Obtain the sum of all balances
40 4 - Obtain the average of all balances
41 5 - Find person
42 0 - Exit
43 Enter an option (0 to exit): 3
44
45 Finding the larger sum…
46
47 MENU OPTIONS
48
49 1 - Find the larger balance
50 2 - Find the smaller balance
51 3 - Obtain the sum of all balances
52 4 - Obtain the average of all balances
53 5 - Find person
54 0 - Exit
55 Enter an option (0 to exit): 4
56
57 Finding the larger average…
```

```
 58
 59 MENU OPTIONS
 60
 61 1 - Find the larger balance
 62 2 - Find the smaller balance
 63 3 - Obtain the sum of all balances
 64 4 - Obtain the average of all balances
 65 5 - Find person
 66 0 - Exit
 67 Enter an option (0 to exit): 5
 68
 69 Who do you want to search for (enter done to exit): Chris Carroll
 70 Found
 71
 72 MENU OPTIONS
 73
 74 1 - Find the larger balance
 75 2 - Find the smaller balance
 76 3 - Obtain the sum of all balances
 77 4 - Obtain the average of all balances
 78 5 - Find person
 79 0 - Exit
 80 Enter an option (0 to exit): 5
 81
 82 Who do you want to search for (enter done to exit): Pete McBride
 83 Found
 84
 85 MENU OPTIONS
 86
 87 1 - Find the larger balance
 88 2 - Find the smaller balance
 89 3 - Obtain the sum of all balances
 90 4 - Obtain the average of all balances
 91 5 - Find person
 92 0 - Exit
 93 Enter an option (0 to exit): 5
 94
 95 Who do you want to search for (enter done to exit): Jean Rousseau
 96 Found
 97
 98 MENU OPTIONS
 99
100 1 - Find the larger balance
101 2 - Find the smaller balance
102 3 - Obtain the sum of all balances
103 4 - Obtain the average of all balances
104 5 - Find person
105 0 - Exit
106 Enter an option (0 to exit): 5
107
108 Who do you want to search for (enter done to exit): Florence Cyr
109 Florence Cyrwas not found.
110
111 MENU OPTIONS
112
113 1 - Find the larger balance
114 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

```

```
 1 ********************************************
 2 * PROGRAMMED BY : Faris Hijazi
 3 * CLASS         : CS1A
 4 * SECTION       : MW: 7:30P
 5 * Assignment #1 : Functions and Arrays
 6 ********************************************
 7
 8 Larger Balance:
 9 ID #     NAME                    BALLANCE DUE
10 ----     --------------------    -----------
11 1002     Steve Woolston          $   1423.20
12
13 Smaller Balance:
14 ID #     NAME                    BALLANCE DUE
15 ----     --------------------    -----------
16 1002     Steve Woolston          $   1423.20
17
18 Sum of Balance for all persons:
19 $   4080.48
20
21 Average of Balance for all persons:
22 $    408.05
23
24 Search Name:
25 ID #     NAME                    BALLANCE DUE
26 ----     --------------------    -----------
27 1008     Chris Carroll           $     32.35
28
29 Search Name:
30 ID #     NAME                    BALLANCE DUE
31 ----     --------------------    -----------
32 1007     Pete McBride            $    500.32
33
34 Search Name:
35 ID #     NAME                    BALLANCE DUE
36 ----     --------------------    -----------
37 1001     Jean Rousseau           $     15.50
38
39
```

```
1 /*****************************************************************************
2  * AUTHOR       : Faris Hijazi
3  * STUDENT ID   : 1039438
4  * ASSIGNMENT 1 : Functions and Arrays
5  * CLASS        : CS1A
6  * SECTION      : MW 7:30PM
7  * DUE DATE     : 02/15/19
8  *****************************************************************************/
9 #include <iostream>
10 #include <iomanip>
11 #include <cstring>
12 #include <sstream>
13 #include <string>
14 #include <fstream>
15 using namespace std;
16
17 //ENUMERATED TYPES
18 enum menuoption {EXIT,
19                  LARGERBALANCE,
20                  SMALLERBALANCE,
21                  SUM,
22                  AVERAGE,
23                  SEARCH};
24
25 //PROTOTYPES
26 void printHeader(ostream &output,      //output device
27                  char exersize,        //lab or assignment?
28                  string exersizeName, //lab or assignment name
29                  int num,              //lab or assignment name
30                  string names          //names of programmer(s)
31                  );
32
33 void input(string &outFileName,       //name of output file
34            string ar1[],              //string array
35            int ar2[],                 //int array
36            float ar3[],               //float array
37            const int AR_SIZE          //size of parallel arrays
38            );
39
40 void menu();
41
42 float sumAvg(menuoption token,        //menu option chosen
43              float ar3[],             //float array
44              const int AR_SIZE        //size of array
45              );
46
47 int searchBalance(menuoption token,   //menu option chosen
48                   float ar3[],        //float array
49                   const int AR_SIZE   //size of array
50                   );
51
52 int search(menuoption token,          //menu option chosen
53            string ar1[],              //string array
54            const int AR_SIZE,         //size of array
55            string userInput,          //user inputed search item
56            bool &found                //returns true if found or false if not found
57            );
```

58

```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;
```

```
58
59              case LARGERBALANCE  : searchIndex = searchBalance(LARGERBALANCE, balance,
    AR_SIZE);
60                                    cout  << "\nFinding the larger balance…\n\n";
61                                    oFile << "Larger Balance:\n";
62                                    oFile << left << setw(9) << "ID #"
63                                          << setw(25) << "NAME" << "BALLANCE DUE\n";
64                                    oFile << left << setw(9) << "----"
65                                          << setw(25) << "--------------------" <<
    "-----------\n";
66                                    oFile << left << setw(9) << id[searchIndex] << setw(25)
67                                          << name[searchIndex];
68                                    oFile <<  "$" << setprecision(2) << right << fixed <<
    setw(10) << balance[searchIndex]
69                                          << endl << endl << setprecision(6);
70                                    oFile.unsetf(ios::fixed);
71                                    break;
72
73              case SMALLERBALANCE : searchIndex = searchBalance(LARGERBALANCE, balance,
    AR_SIZE);
74                                    cout  << "\nFinding the smaller balance…\n";
75                                    oFile << "Smaller Balance:\n";
76                                    oFile << left << setw(9) << "ID #"
77                                          << setw(25) << "NAME" << "BALLANCE DUE\n";
78                                    oFile << left << setw(9) << "----"
79                                          << setw(25) << "--------------------" <<
    "-----------\n";
80                                    oFile << left << setw(9) << id[searchIndex] << setw(25)
81                                          << name[searchIndex];
82                                    oFile <<  "$" << setprecision(2) << right << fixed <<
    setw(10) << balance[searchIndex]
83                                          << endl << endl << setprecision(6);
84                                    oFile.unsetf(ios::fixed);
85                                    break;
86
87              case SUM            : cout  << "\nFinding the larger sum…\n";
88                                    oFile << "Sum of Balance for all persons:\n";
89                                    oFile << setprecision(2) << fixed;
90                                    oFile << "$" << setw(10) << sumAvg(SUM, balance, AR_SIZE);
91                                    oFile << endl << endl;
92                                    oFile << setprecision(6);
93                                    oFile.unsetf(ios::fixed);
94                                    break;
95
96              case AVERAGE        : cout  << "\nFinding the larger average…\n";
97                                    oFile << "Average of Balance for all persons:\n";
98                                    oFile << setprecision(2) << fixed;
99                                    oFile << "$" << setw(10) << sumAvg(AVERAGE, balance,
    AR_SIZE);
100                                   oFile << endl << endl;
101                                   oFile << setprecision(6);
102                                   oFile.unsetf(ios::fixed);
103                                   break;
104
105             case SEARCH         : cout << "\nWho do you want to search for (enter done to
    exit): ";
106                                   getline(cin, nameSearch);
```

```
107
108                                     searchIndex = search(LARGERBALANCE, name, AR_SIZE,
    nameSearch, nameFound);
109
110                                     if(nameFound)
111                                     {
112                                         oFile << "Search Name:\n";
113                                         oFile << left << setw(9) << "ID #"
114                                               <<setw(25) << "NAME" << "BALLANCE DUE\n";
115                                         oFile << left << setw(9) << "----"
116                                               <<setw(25) << "--------------------" <<
    "-----------\n";
117                                         oFile << left << setw(9) << id[searchIndex] << setw(25)
118                                               << name[searchIndex];
119                                         oFile <<  "$" << setprecision(2) << right << fixed <<
    setw(10) << balance[searchIndex]
120                                               << endl << endl << setprecision(6);
121                                         oFile.unsetf(ios::fixed);
122                                      }
123                                     break;
124             }
125             menu();
126             cin >> menuInput;
127             cin.ignore(1000, '\n');
128             menuChoice = menuoption(menuInput);
129         }
130         return(0);
131 }
132
```

```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);
}

```

```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);
}

```

```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 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);
    }
}
```

```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 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();
}


```

```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;
}

```

```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): ";
}

```