```
*******************************************************
*   PROGRAMMED BY : Carlos Aguilera
*   CLASS         : CS1B
*   SECTION       : MW: 7:30p - 9:50p
*   LAB #1        : Functions & 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...

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

Obtaining the sum of all Balances...

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

Obtaining the average of all Balances...

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): Steve Woolston
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): Jacques Rousseau
Jacques Rousseau was 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): 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
```

```
120
121 Who do you want to search for (enter done to exit): Jean Rousseau
122 Found.
123
124 Menu Options
125
126 1 - Find the larger balance
127 2 - Find the smaller balance
128 3 - Obtain the sum of all balances
129 4 - Obtain the average of all balances
130 5 - Find Person
131 0 - Exit
132 Enter an option (0 to exit): 5
133
134 Who do you want to search for (enter done to exit): Florence Cyr
135 Florence Cyr was not found.
136
137 Menu Options
138
139 1 - Find the larger balance
140 2 - Find the smaller balance
141 3 - Obtain the sum of all balances
142 4 - Obtain the average of all balances
143 5 - Find Person
144 0 - Exit
145 Enter an option (0 to exit): 0
146
147 Thank you for using my program.
```

```
Larger Balance:
ID #     NAME                    BALANCE DUE
----     --------------------    -----------
1002     Steve Woolston          $    1423.2

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

Sum of Balance for all persons:
$   4080.48

Average Balance for all persons:
$    408.05

Search Name:
ID #     NAME                    BALANCE DUE
----     --------------------    -----------
1002     Steve Woolston          $    1423.2

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           $      15.5


```

```
#pragma once

#include <iostream>
#include <iomanip>
#include <string>
#include <fstream>

void heading();
void readFile(std::string inputFileName, size_t sizeofArray, std::string arrayofNames[], int arrayofIDs[], double arrayofBalances[]);
int balanceIndex(char selection, size_t sizeofArray, double arrayofBalances[]);
double sumofBalances(size_t sizeofArray, double arrayofBalances[]);
int searchName(std::string inputName, size_t sizeofArray, std::string arrayofNames[]);
```

```cpp
/*****************************************************************************
*
 * AUTHOR      : Carlos Aguilera
 * STUDENT ID  : 1152562
 * LAB #       : 1
 * CLASS       : CS1B
 * SECTION     : M-W
 * DUE DATE    : 02.02.22
 *****************************************************************************
/

#include "main.hpp"

/*****************************************************************************
 * Title: Functions & Arrays
 * ---------------------------------------------------------------------------
 * This program will output the class heading
 * ---------------------------------------------------------------------------
 * INPUT:
 * inputFileName {file name for input}
 * outputFileName {file name for output}
 * selection {char for switch case condition}
 * inputName {name input to search for from user}
 * inFile {read from input file}
 * OUTPUT:
 * outFile {output to file}
 *****************************************************************************/

int main()
{
  heading();

  std::string inputFileName {}, outputFileName {}, temp {};
  std::cout << "What input file would you like to use? ";
  std::cin >> inputFileName; //reads input for what file to read from
  std::cout << "What output file would you like to use? ";
  std::cin >> outputFileName; // reads input for what file to write to

  size_t sizeofArray {0};
  std::fstream inFile;
  inFile.open(inputFileName, std::ios::in); //file is in read only mode
  while(getline(inFile, temp))//stores line in temporary string
    ++sizeofArray;// a loop that gets the number of lines in the file
  sizeofArray /= 2;// works for the type of formatting that the input file
has if format changes then bugs could occur
  inFile.close();

  std::string arrayofNames[sizeofArray];
  int arrayofIDs[sizeofArray];
  double arrayofBalances[sizeofArray];

  readFile(inputFileName, sizeofArray, arrayofNames, arrayofIDs,
arrayofBalances);//reads file and sets values in the arrays

  char selection {};
  do
  {
    std::cout << "\nMenu Options\n\n"
        << "1 - Find the larger balance\n"
```

```
56          << "2 - Find the smaller balance\n"
57          << "3 - Obtain the sum of all balances\n"
58          << "4 - Obtain the average of all balances\n"
59          << "5 - Find Person\n"
60          << "0 - Exit\n"
61              << "Enter an option (0 to exit): ";
62      std::cin >> selection;
63      std::cout << "\n";
64
65      switch (selection)
66      {
67        case '1': {
68          std::cout << "Finding the Larger Balance...\n";
69
70          std::fstream outFile;
71          outFile.open(outputFileName, std::ios::app);//appends to file and
    doesn't erase but adds instead
72          outFile << "Larger Balance:\n";
73          outFile << "ID #     NAME                     BALANCE DUE\n";
74          outFile << "----     --------------------     -----------\n";
75          outFile << arrayofIDs[balanceIndex(selection, sizeofArray,
    arrayofBalances)] << "     ";//returns an index for the largest balance in
    the input file
76          outFile << arrayofNames[balanceIndex(selection, sizeofArray,
    arrayofBalances)];
77          outFile << std::setw(26 - arrayofNames[balanceIndex(selection,
    sizeofArray, arrayofBalances)].size());//returns the size of the largest
    balance name and subtracts a set width of 26 to get proper format
78          outFile << "$" << std::setw(10) <<
    arrayofBalances[balanceIndex(selection, sizeofArray, arrayofBalances)] <<
    "\n\n";
79          outFile.close();
80          break;
81        }
82        case '2': {
83          std::cout << "Finding the Smaller Balance...\n";
84
85          std::fstream outFile;
86          outFile.open(outputFileName, std::ios::app);
87          outFile << "Smaller Balance:\n";
88          outFile << "ID #     NAME                     BALANCE DUE\n";
89          outFile << "----     --------------------     -----------\n";
90          outFile << arrayofIDs[balanceIndex(selection, sizeofArray,
    arrayofBalances)] << "     ";
91          outFile << arrayofNames[balanceIndex(selection, sizeofArray,
    arrayofBalances)];
92          outFile << std::setw(26 - arrayofNames[balanceIndex(selection,
    sizeofArray, arrayofBalances)].size());
93          outFile << "$" << std::setw(10) <<
    arrayofBalances[balanceIndex(selection, sizeofArray, arrayofBalances)] <<
    "\n\n";
94          outFile.close();
95          break;
96        }
97        case '3': {
98          std::cout << "Obtaining the sum of all Balances...\n";
99
100         std::fstream outFile;
101         outFile.open(outputFileName, std::ios::app);
102         outFile << "Sum of Balance for all persons:\n";
```

```
103         outFile << std::fixed << std::setprecision(2) << "$" << std::setw(10)
    << sumofBalances(sizeofArray, arrayofBalances) << "\n\n";//returns sum of
    balances
104         outFile.close();
105         break;
106       }
107       case '4': {
108         std::cout << "Obtaining the average of all Balances...\n";
109 
110         std::fstream outFile;
111         outFile.open(outputFileName, std::ios::app);
112         outFile << "Average Balance for all persons:\n";
113         outFile << std::fixed << std::setprecision(2) << "$" << std::setw(10)
    << sumofBalances(sizeofArray, arrayofBalances)/sizeofArray <<
    "\n\n";//returns average of balances using the size of array or how many IDs
    we have
114         outFile.close();
115         break;
116       }
117       case '5': {
118         std::string inputName {};
119         std::cout << "Who do you want to search for (enter done to exit): ";
120         std::cin.ignore(10, '\n');
121         std::getline(std::cin, inputName);
122 
123         if(inputName == "done")//expection handling for when user enters done
124           continue;
125         else if(searchName(inputName, sizeofArray, arrayofNames) != -1) {//
    function returns a -1 if not found and if found returns index that it was
    found in
126           std::cout << "Found.\n";
127 
128           std::fstream outFile;
129           outFile.open(outputFileName, std::ios::app);
130           outFile << "Search Name:\n";
131           outFile << "ID #     NAME                    BALANCE DUE\n";
132           outFile << "----     --------------------    -----------\n";
133           outFile << arrayofIDs[searchName(inputName, sizeofArray,
    arrayofNames)] << "     ";
134           outFile << arrayofNames[searchName(inputName, sizeofArray,
    arrayofNames)];
135           outFile << std::setw(26 - arrayofNames[searchName(inputName,
    sizeofArray, arrayofNames)].size());
136           outFile << "$" << std::setw(10) <<
    arrayofBalances[searchName(inputName, sizeofArray, arrayofNames)] << "\n\n";
137           outFile.close();
138         }else
139           std::cout << inputName << " was not found.\n";//handling not found
140         break;
141       }
142       case '0': {
143         std::cout << "Thank you for using my program.\n";
144         break;
145       }
146       default: {
147         std::cout << "Invalid input!\n";
148         break;
149       }
150     }
151   } while (selection != '0');
```

```
152
153 }
```

```cpp
#include "main.hpp"

void readFile(std::string inputFileName, size_t sizeofArray, std::string arrayofNames[], int arrayofIDs[], double arrayofBalances[])
{
  std::string fname {}, lname {};
  std::fstream inFile;
  inFile.open(inputFileName, std::ios::in);//read mode
  for(size_t i {0}; i < sizeofArray; i++)
  {
    inFile >> fname >> lname;//wanted to use getline but then thought this could be less error prone
    arrayofNames[i] = fname + ' ' + lname;//concat fname and lname and assigning it to array names at index i
    inFile >> arrayofIDs[i] >> arrayofBalances[i];// reading id and balance
  }
  inFile.close();
}
```

```cpp
#include "main.hpp"

int balanceIndex(char selection, size_t sizeofArray, double arrayofBalances[])
{
  if(selection == '1')//if user chose larger in the menu selection
  {
    int index {};
    double balance {arrayofBalances[0]};//did not want to use a nested for loop so initialized balance to index 0 of arrayofBalances
    for(size_t i {1}; i < sizeofArray; i++)
    {
      if(arrayofBalances[i] > balance)//if array of balances at index i is greater than balance(because we want the largest) then assign it to balance and assign i to index
      {
        balance = arrayofBalances[i];
        index = i;
      }
    }
    return index;
  }else {
    int index {};
    double balance {arrayofBalances[0]};
    for(size_t i {1}; i < sizeofArray; i++)
    {
      if(arrayofBalances[i] < balance)//same thing but for smallest value
      {
        balance = arrayofBalances[i];
        index = i;
      }
    }
    return index;
  }
}
```

```cpp
#include "main.hpp"

double sumofBalances(size_t sizeofArray, double arrayofBalances[])
{
  double balanceSum {};
  for(size_t i {0}; i < sizeofArray; i++)
    balanceSum += arrayofBalances[i];//takes in balance for each iteration of the for loop and plus equals it
  return balanceSum;//returns result
}
```

```cpp
#include "main.hpp"

int searchName(std::string inputName, size_t sizeofArray, std::string arrayofNames[])
{
    for(size_t i {0}; i < sizeofArray; i++)
    {
        if(inputName == arrayofNames[i])//checking if the name user inputed exists
            return i;// returns index if true
    }
    return -1;//returns -1 if not found
}
```

```cpp
#include "main.hpp"

void heading()
{
    /*****************************************************************************
     * 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
     *****************************************************************************/
  const char PROGRAMMER[] = "Carlos Aguilera";
  const char CLASS[]      = "CS1B";
  const char SECTION[]    = "MW: 7:30p - 9:50p";
  const int LAB_NUM       = 1;
  const char LAB_NAME[]   = "Functions & Arrays";

  /*****************************************************************************
   * OUTPUT - Class Heading
   *****************************************************************************/
  std::cout << std::left;
  std::cout << "*************************************************\n";
  std::cout << "*   PROGRAMMED BY : " << PROGRAMMER << std::endl;
  std::cout << "*   " << std::setw(14) <<"CLASS" << ": " << CLASS <<
std::endl;
  std::cout << "*   " << std::setw(14) <<"SECTION" << ": " << SECTION <<
std::endl;
  std::cout << "*   LAB #" << std::setw(9) << LAB_NUM << ": " << LAB_NAME <<
std::endl;
  std::cout << "*************************************************\n\n";
  std::cout << std::right;
  /*****************************************************************************/
}
```

```
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
```