Name: Garrett Jackson

Vectors and Pointers

To be reported on canvas. Create a PDF. Include screenshots of code and execution. Include copy-pasteable text of code. Be careful with variable names and indentation. You must use the templates.

Problem 3. Number Analysis Program

Write a program that asks the user for a file name. Assume the file contains a series of integer numbers, each written on a separate line (maximum 20). The program should read the contents of the file into an array and then display the following data:

- The lowest number in the array
- The highest number in the area
- The total of the numbers in the array
- The average of the numbers in the array

TEMPLATE:

```
//DO NOT MODIFY THIS SECTION
#include <iostream>
#include <fstream>
using namespace std;
const int SIZE=50;
//prototypes (INSERT HERE IF YOU USE FUNCTIONS)

//end prototypes
int main()
{
    ifstream ifile;
    string fileName;
    int lowest, highest, total=0, counter=0;

    int numbers[SIZE];
    double average;
    cout << "Name of file: ";
    cin >> fileName;
    ifile.open( fileName );
    if( ifile.fail() )
    {
        cout << "Error" << endl;
        return 1;
    }
//ADD YOUR CODE FROM HERE
```

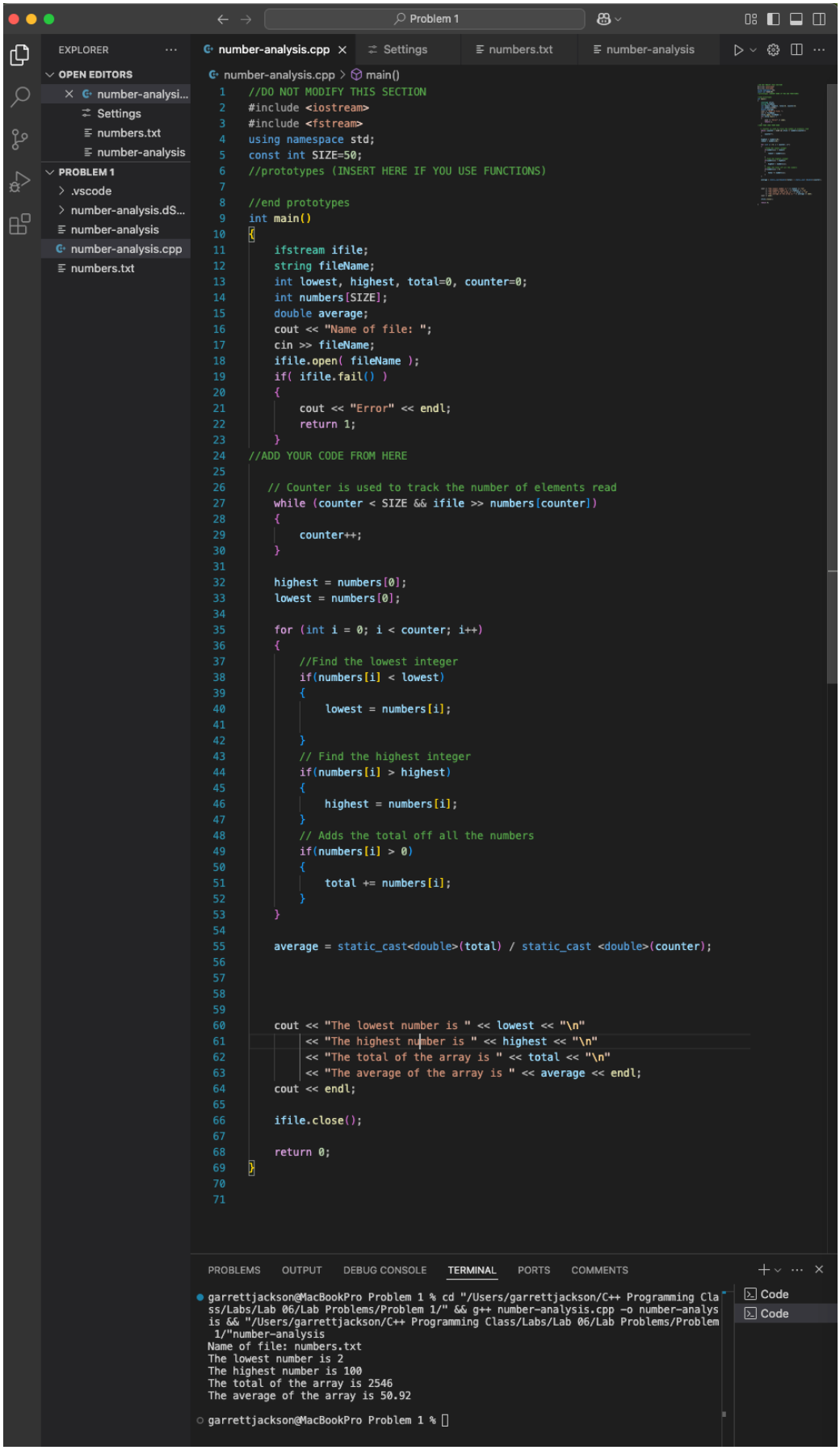
Execution:

```
Name of file: numbers.txt
Lowest number: 12
Highest number: 123
```

```
Total: 461
Average: 57.625
```

File "numbers.txt" for the example:

```
17
12
45
78
34
123
98
54
```


Problem 1
EXPLORER
OPEN EDITORS
number-analysi...
Settings
numbers.txt
number-analysis
PROBLEM 1
.vscode
number-analysis.dS...
number-analysis
number-analysis.cpp
numbers.txt
number-analysis.cpp
Settings
numbers.txt
number-analysis
number-analysis.cpp > main()
//DO NOT MODIFY THIS SECTION
#include <iostream>
#include <fstream>
using namespace std;
const int SIZE=50;
//prototypes (INSERT HERE IF YOU USE FUNCTIONS)
//end prototypes
int main()
{
ifstream ifile;
string fileName;
int lowest, highest, total=0, counter=0;
int numbers[SIZE];
double average;
cout << "Name of file: ";
cin >> fileName;
ifile.open(fileName);
if(ifile.fail())
{
cout << "Error" << endl;
return 1;
}
//ADD YOUR CODE FROM HERE
// Counter is used to track the number of elements read
while (counter < SIZE && ifile >> numbers[counter])
{
counter++;
}
highest = numbers[0];
lowest = numbers[0];
for (int i = 0; i < counter; i++)
{
//Find the lowest integer
if(numbers[i] < lowest)
{
lowest = numbers[i];
}
// Find the highest integer
if(numbers[i] > highest)
{
highest = numbers[i];
}
// Adds the total off all the numbers
if(numbers[i] > 0)
{
total += numbers[i];
}
}
average = static_cast<double>(total) / static_cast <double>(counter);
cout << "The lowest number is " << lowest << "\n"
<< "The highest number is " << highest << "\n"
<< "The total of the array is " << total << "\n"
<< "The average of the array is " << average << endl;
cout << endl;
ifile.close();
return 0;
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
garrettjackson@MacBookPro Problem 1 % cd "/Users/garrettjackson/C++ Programming Class/Labs/Lab 06/Lab Problems/Problem 1/" && g++ number-analysis.cpp -o number-analysis && "/Users/garrettjackson/C++ Programming Class/Labs/Lab 06/Lab Problems/Problem 1/"number-analysis
Name of file: numbers.txt
The lowest number is 2
The highest number is 100
The total of the array is 2546
The average of the array is 50.92
garrettjackson@MacBookPro Problem 1 %
Code
Code

numbers.txt

```
27
84
13
59
92
6
75
48
31
97
42
88
19
53
70
4
95
36
11
67
81
23
90
14
58
79
2
61
99
33
8
47
29
100
55
21
73
43
96
16
39
62
5
87
30
77
12
98
66
25
```

```
//DO NOT MODIFY THIS SECTION
#include <iostream>
#include <fstream>
using namespace std;
const int SIZE=50;
//prototypes (INSERT HERE IF YOU USE FUNCTIONS)

//end prototypes
int main()
{
    ifstream ifile;
    string fileName;
    int lowest, highest, total=0, counter=0;
    int numbers[SIZE];
    double average;
    cout << "Name of file: ";
    cin >> fileName;
    ifile.open( fileName );
    if( ifile.fail() )
    {
        cout << "Error" << endl;
        return 1;
    }
//ADD YOUR CODE FROM HERE

   // Counter is used to track the number of elements read
    while (counter < SIZE && ifile >> numbers[counter])
    {
        counter++;
    }

    highest = numbers[0];
    lowest = numbers[0];

    for (int i = 0; i < counter; i++)
    {
        //Find the lowest integer
        if(numbers[i] < lowest)
        {
            lowest = numbers[i];

        }
        // Find the highest integer
        if(numbers[i] > highest)
        {
            highest = numbers[i];
        }
        // Adds the total off all the numbers
        if(numbers[i] > 0)
        {
            total += numbers[i];
        }
    }

    average = static_cast<double>(total) / static_cast <double>(counter);
```

```
    cout << "The lowest number is " << lowest << "\n"
         << "The highest number is " << highest << "\n"
         << "The total of the array is " << total << "\n"
         << "The average of the array is " << average << endl;
    cout << endl;

    ifile.close();

    return 0;
}
```

Problem 5. Menu Driven Program Statistics

From Problem 2, add an option to display the next information:

- Older student (name)
- Young student (name)
- Age average.

menu-highandlow.cpp

```cpp
#include <iostream>
#include <fstream>
using namespace std;

const int MAX_ENTRIES = 50; // Maximum allowed entries

int main()
{
    // 1. Declare variables and arrays
    int ages[MAX_ENTRIES];
    string names[MAX_ENTRIES];
    int highestAge, lowestAge, totalAges, index;
    string highestName, lowestName;
    double average;
    char option;
    int numEntries = 0; // Track the number of entries
    fstream ifs;
    ofstream ofs;

    // 2. Open and read the file
    ifs.open("student.txt");

    if (!ifs.is_open())
    {
        cout << "File not found. A new file will be created when saving
    }
    else
    {
        while (numEntries < MAX_ENTRIES && ifs >> ages[numEntries])
        {
            ifs.ignore();
            getline(ifs, names[numEntries]);
            numEntries++;
        }
        ifs.close();
    }

    // 3. Menu loop
    while (true)
    {
        // Display Menu
        cout << "\nMenu:\n"
             << "1: Modify an Entry\n"
             << "2: Display Data\n"
             << "3: Add New Entry\n"
             << "4: Save Data\n"
             << "5: Show Oldest, Youngest, and Average Age\n"
             << "6: Quit the Program\n"
             << "Option: ";
        cin >> option;
        cin.ignore();

        switch (option)
        {
            case '1': // Modify an Entry
                cout << "Enter entry number (1-" << numEntries << ") to
                cin >> index;
                cin.ignore();

                if (index >= 1 && index <= numEntries)
                {
                    cout << "Enter a new age: ";
                    cin >> ages[index - 1];
                    cin.ignore();
                    cout << "Enter a new name: ";
                    getline(cin, names[index - 1]);
                }
                else
                {
                    cout << "Invalid entry number.\n";
                }
                break;

            case '2': // Display Data
                if (numEntries == 0)
                {
                    cout << "No data available.\n";
                }
                else
                {
                    for (int i = 0; i < numEntries; i++)
                    {
                        cout << i + 1 << ": " << ages[i] << " - " << na
                    }
                }
                break;

            case '3': // Add New Entry

                cout << "Enter 'Done' in the name field when finished"
                while(numEntries < MAX_ENTRIES)
                {

                    cout << "Enter age: ";
                    cin >> ages[numEntries];
                    cin.ignore();
                    if(ages[numEntries] )
                    cout << "Enter name: ";
                    getline(cin, names[numEntries]);

```

```
garrettjackson@Garretts-MacBook-Pro Problem 2 % cd "/
Users/gar
rettjackson/C++ Programming Cla
ss/Labs/Lab 06/Lab Problems/Pro
blem 2/" && g++ menu-highandlow
.cpp -o menu-highandlow && "/Us
ers/garrettjackson/C++ Programm
ing Class/Labs/Lab 06/Lab Probl
ems/Problem 2/"menu-highandlow

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Garrett
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 1
Enter entry number (1-8) to modify: 1
Enter a new age: 24
Enter a new name: Ace

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Ace
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 3
Enter 'Done' in the name field when finished

Enter age: 26
Enter name: Kenny
Enter age: 0
1
Enter age: 23
Enter name: done
Exiting entry mode.

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Ace
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer
9: 26 - Kenny

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 4
Data saved successfully!

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 5

The oldest students age: 29( Travis )
The youngest students age: 19( Trinidy )
Average age: 21

Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 6
garrettjackson@Garretts-MacBook-Pro Problem 2 %
```


Problem 2
menu-highandlow.cpp
menu-highandlow.cpp > main()
int main()
while (true)
switch (option)
while(numEntries < MAX_ENTRIES)
// Check if user wants to exit
if (names[numEntries] == "Done" || names[numEntries
{
cout << "Exiting entry mode.\n";
numEntries--;
break;
}
numEntries++;
}
if(numEntries >= MAX_ENTRIES)
{
cout << "Maximum entries reached. Cannot add more.\
break;
}
continue;
case '4': // Save Data
ofs.open("student.txt"); // Overwrites the file with up
if (!ofs)
{
cout << "Error opening file for writing!\n";
}
else
{
for (int i = 0; i < numEntries; i++)
{
ofs << ages[i] << endl << names[i] << endl;
}
ofs.close();
cout << "Data saved successfully!\n";
}
break;
case '5': // Oldest, Youngest, and Average
highestAge = ages[0];
lowestAge = ages[0];
highestName = names[0];
lowestName = names[0];
totalAges = 0;
for(int i = 1; i < numEntries; i++)
{
// Finds the lowest Age of a Student
if(ages[i] < lowestAge)
{
lowestAge = ages[i];
lowestName = names[i];
}
// Finds the highest Age of the students
if(ages[i] > highestAge)
{
highestAge = ages[i];
highestName = names[i];
}
totalAges += ages[i];
}
//Averages the ages and returns a double
average = totalAges / numEntries;
//Display the new data
cout << "\n"
<< "The oldest students age: " << highestAge << "(" <
<< "The youngest students age: " << lowestAge << "("
<< "Average age: " << average << endl;
break;
case '6': //Quits the Program
return 0;
default: //Input Validation
cout << "Invalid option. Please try again.\n";
}
}
}
PROBLEMS
TERMINAL
Code
garrettjackson@Garretts-MacBook-Pro Problem 2 % cd "/
Users/gar
rettjackson/C++ Programming Cla
ss/Labs/Lab 06/Lab Problems/Pro
blem 2/" && g++ menu-highandlow
.cpp -o menu-highandlow && "/Us
ers/garrettjackson/C++ Programm
ing Class/Labs/Lab 06/Lab Probl
ems/Problem 2/"menu-highandlow
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Garrett
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 1
Enter entry number (1-8) to modify: 1
Enter a new age: 24
Enter a new name: Ace
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Ace
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 3
Enter 'Done' in the name field when finished
Enter age: 26
Enter name: Kenny
Enter age: 0
1
Enter age: 23
Enter name: done
Exiting entry mode.
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 2
1: 24 - Ace
2: 19 - Trinidy
3: 24 - Rosie
4: 23 - Ethan
5: 23 - Dohn
6: 22 - Allissa
7: 29 - Travis
8: 27 - Tracer
9: 26 - Kenny
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 4
Data saved successfully!
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 5
The oldest students age: 29(Travis)
The youngest students age: 19(Trinidy)
Average age: 21
Menu:
1: Modify an Entry
2: Display Data
3: Add New Entry
4: Save Data
5: Show Oldest, Youngest, and Average Age
6: Quit the Program
Option: 6
garrettjackson@Garretts-MacBook-Pro Problem 2 %

```
#include <iostream>
#include <fstream>
using namespace std;

const int MAX_ENTRIES = 50; // Maximum allowed entries

int main()
{
   // 1. Declare variables and arrays
   int ages[MAX_ENTRIES];
   string names[MAX_ENTRIES];
   int highestAge, lowestAge, totalAges, index;
   string highestName, lowestName;
   double average;
   char option;
   int numEntries = 0; // Track the number of entries
   fstream ifs;
   ofstream ofs;

   // 2. Open and read the file
   ifs.open("student.txt");

   if (!ifs.is_open())
   {
      cout << "File not found. A new file will be created when saving.\n";
   }
   else
   {
      while (numEntries < MAX_ENTRIES && ifs >> ages[numEntries])
      {
         ifs.ignore();
         getline(ifs, names[numEntries]);
         numEntries++;
      }
      ifs.close();
   }

   // 3. Menu loop
   while (true)
   {
      // Display Menu
      cout << "\nMenu:\n"
           << "1: Modify an Entry\n"
           << "2: Display Data\n"
           << "3: Add New Entry\n"
           << "4: Save Data\n"
           << "5: Show Oldest, Youngest, and Average Age\n"
           << "6: Quit the Program\n"
```

```
     << "Option: ";
cin >> option;
cin.ignore();

switch (option)
{
   case '1': // Modify an Entry
      cout << "Enter entry number (1-" << numEntries << ") to modify: ";
      cin >> index;
      cin.ignore();

      if (index >= 1 && index <= numEntries)
      {
         cout << "Enter a new age: ";
         cin >> ages[index - 1];
         cin.ignore();
         cout << "Enter a new name: ";
         getline(cin, names[index - 1]);
      }
      else
      {
         cout << "Invalid entry number.\n";
      }
      break;

   case '2': // Display Data
      if (numEntries == 0)
      {
         cout << "No data available.\n";
      }
      else
      {
         for (int i = 0; i < numEntries; i++)
         {
            cout << i + 1 << ": " << ages[i] << " - " << names[i] << endl;
         }
      }
      break;

   case '3': // Add New Entry

      cout << "Enter 'Done' in the name field when finished" << endl << endl;
      while(numEntries < MAX_ENTRIES)
      {

         cout << "Enter age: ";
         cin >> ages[numEntries];
         cin.ignore();
```

```
            if(ages[numEntries] )
            cout << "Enter name: ";
            getline(cin, names[numEntries]);

            // Check if user wants to exit
            if (names[numEntries] == "Done" || names[numEntries] == "done")
            {
               cout << "Exiting entry mode.\n";
               numEntries--;
               break;
            }
            numEntries++;
         }
         if(numEntries >= MAX_ENTRIES)
         {
            cout << "Maximum entries reached. Cannot add more.\n";
            break;
         }
         continue;

      case '4': // Save Data
         ofs.open("student.txt"); // Overwrites the file with updated data
         if (!ofs)
         {
            cout << "Error opening file for writing!\n";
         }
         else
         {
            for (int i = 0; i < numEntries; i++)
            {
               ofs << ages[i] << endl << names[i] << endl;
            }
            ofs.close();
            cout << "Data saved successfully!\n";
         }
         break;

      case '5': // Oldest, Youngest, and Average
      highestAge = ages[0];
      lowestAge = ages[0];
      highestName = names[0];
      lowestName = names[0];
      totalAges = 0;

      for(int i = 1; i < numEntries; i++)
      {
         // Finds the lowest Age of a Student
         if(ages[i] < lowestAge)
```

```
            {
               lowestAge = ages[i];
               lowestName = names[i];
            }
            // Finds the highest Age of the students
            if(ages[i] > highestAge)
            {
               highestAge = ages[i];
               highestName = names[i];
            }

            totalAges += ages[i];
         }

         //Averages the ages and returns a double

         average = totalAges / numEntries;

         //Display the new data
         cout << "\n"
              << "The oldest students age: " << highestAge << "( " << highestName << " )\n"
              << "The youngest students age: " << lowestAge << "( " << lowestName << " )\n"
              << "Average age: " << average << endl;

            break;

         case '6': //Quits the Program

            return 0;

         default: //Input Validation
            cout << "Invalid option. Please try again.\n";
      }

   }

}
```