

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

```
1 //DO NOT MODIFY THIS SECTION
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5 const int SIZE=50;
6 //prototypes (INSERT HERE IF YOU USE FUNCTIONS)
7
8 //end prototypes
9 int main()
10 {
11     ifstream ifile;
12     string fileName;
13     int lowest, highest, total=0, counter=0;
14     int numbers[SIZE];
15     double average;
16     cout << "Name of file: ";
17     cin >> fileName;
18     ifile.open( fileName );
19     if( ifile.fail() )
20     {
21         cout << "Error" << endl;
22         return 1;
23     }
24     //ADD YOUR CODE FROM HERE
25
26     // Counter is used to track the number of elements read
27     while (counter < SIZE && ifile >> numbers[counter])
28     {
29         counter++;
30     }
31
32     highest = numbers[0];
33     lowest = numbers[0];
34
35     for (int i = 0; i < counter; i++)
36     {
37         //Find the lowest integer
38         if(numbers[i] < lowest)
39         {
40             lowest = numbers[i];
41         }
42         // Find the highest integer
43         if(numbers[i] > highest)
44         {
45             highest = numbers[i];
46         }
47         // Adds the total off all the numbers
48         if(numbers[i] > 0)
49         {
50             total += numbers[i];
51         }
52     }
53
54     average = static_cast<double>(total) / static_cast<double>(counter);
55
56     cout << "The lowest number is " << lowest << "\n"
57         << "The highest number is " << highest << "\n"
58         << "The total of the array is " << total << "\n"
59         << "The average of the array is " << average << endl;
60     cout << endl;
61     ifile.close();
62     return 0;
63 }
```

garrettjackson@MacBookPro Problem 1 % cd "/Users/garrettjackson/C++ Programming Class/Labs/Lab 06/Lab Problems/Problem 1/" && g++ number-analysis.cpp -o number-analysis 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 %

```
≡ numbers.txt
```

```
1 27
2 84
3 13
4 59
5 92
6 6
7 75
8 48
9 31
10 97
11 42
12 88
13 19
14 53
15 70
16 4
17 95
18 36
19 11
20 67
21 81
22 23
23 90
24 14
25 58
26 79
27 2
28 61
29 99
30 33
31 8
32 47
33 29
34 100
35 55
36 21
37 73
38 43
39 96
40 16
41 39
42 62
43 5
44 87
45 30
46 77
47 12
48 98
49 66
50 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.

```

C: menu-highandlow.cpp x
C: menu-highandlow.cpp > main()
1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4
5 const int MAX_ENTRIES = 50; // Maximum allowed entries
6
7 int main()
8 {
9     // 1. Declare variables and arrays
10    int ages[MAX_ENTRIES];
11    string names[MAX_ENTRIES];
12    int highestAge, lowestAge, totalAges, index;
13    string highestName, lowestName;
14    double average;
15    char option;
16    int numEntries = 0; // Track the number of entries
17    fstream ifs;
18    ofstream ofs;
19
20    // 2. Open and read the file
21    ifs.open("student.txt");
22
23    if (!ifs.is_open())
24    {
25        cout << "File not found. A new file will be created when saving\n";
26    }
27    else
28    {
29        while (numEntries < MAX_ENTRIES && ifs >> ages[numEntries])
30        {
31            ifs.ignore();
32            getline(ifs, names[numEntries]);
33            numEntries++;
34        }
35        ifs.close();
36    }
37
38    // 3. Menu loop
39    while (true)
40    {
41        // Display Menu
42        cout << "\nMenu:\n";
43        cout << "1: Modify an Entry\n";
44        cout << "2: Display Data\n";
45        cout << "3: Add New Entry\n";
46        cout << "4: Save Data\n";
47        cout << "5: Show Oldest, Youngest, and Average Age\n";
48        cout << "6: Quit the Program\n";
49        cout << "Option: ";
50        cin >> option;
51        cin.ignore();
52
53        switch (option)
54        {
55            case '1': // Modify an Entry
56                cout << "Enter entry number (1-" << numEntries << ") to modify: ";
57                cin >> index;
58                cin.ignore();
59
60                if (index >= 1 && index <= numEntries)
61                {
62                    cout << "Enter a new age: ";
63                    cin >> ages[index - 1];
64                    cin.ignore();
65                    cout << "Enter a new name: ";
66                    getline(cin, names[index - 1]);
67                }
68                else
69                {
70                    cout << "Invalid entry number.\n";
71                }
72                break;
73
74            case '2': // Display Data
75                if (numEntries == 0)
76                {
77                    cout << "No data available.\n";
78                }
79                else
80                {
81                    for (int i = 0; i < numEntries; i++)
82                    {
83                        cout << i + 1 << ": " << ages[i] << " - " << names[i] << "\n";
84                    }
85                }
86                break;
87
88            case '3': // Add New Entry
89
90                cout << "Enter 'Done' in the name field when finished"
91                while(numEntries < MAX_ENTRIES)
92                {
93
94                    cout << "Enter age: ";
95                    cin >> ages[numEntries];
96                    cin.ignore();
97                    if(ages[numEntries] )
98                    cout << "Enter name: ";
99                    getline(cin, names[numEntries]);
100

```

```

garrettjackson@Garretts-MacBook-Pro Problem 2 % cd "/Users/garrettjackson/C++ Programming Class/Labs/Lab 06/Lab Problems/Problem 2/" && g++ menu-highandlow.cpp -o menu-highandlow && ./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 - Trinity
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 - Trinity
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 - Trinity
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( Trinity )
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

```

```

C- menu-highandlow.cpp x
C- menu-highandlow.cpp > main()
7 int main()
39 while (true)
53 switch (option)
91 while(numEntries < MAX_ENTRIES)
  getLine(cin, names[numEntries]);
100
101 // Check if user wants to exit
102 if (names[numEntries] == "Done" || names[numEntries]
103 {
104     cout << "Exiting entry mode.\n";
105     numEntries--;
106     break;
107 }
108 numEntries++;
109 }
110 if(numEntries >= MAX_ENTRIES)
111 {
112     cout << "Maximum entries reached. Cannot add more.\n";
113     break;
114 }
115 continue;
116
117 case '4': // Save Data
118     ofs.open("student.txt"); // Overwrites the file with up
119     if (!ofs)
120     {
121         cout << "Error opening file for writing!\n";
122     }
123     else
124     {
125         for (int i = 0; i < numEntries; i++)
126         {
127             ofs << ages[i] << endl << names[i] << endl;
128         }
129         ofs.close();
130         cout << "Data saved successfully!\n";
131     }
132     break;
133
134 case '5': // Oldest, Youngest, and Average
135     highestAge = ages[0];
136     lowestAge = ages[0];
137     highestName = names[0];
138     lowestName = names[0];
139     totalAges = 0;
140
141     for(int i = 1; i < numEntries; i++)
142     {
143         // Finds the lowest Age of a Student
144         if(ages[i] < lowestAge)
145         {
146             lowestAge = ages[i];
147             lowestName = names[i];
148         }
149         // Finds the highest Age of the students
150         if(ages[i] > highestAge)
151         {
152             highestAge = ages[i];
153             highestName = names[i];
154         }
155     }
156     totalAges += ages[i];
157 }
158
159 //Averages the ages and returns a double
160
161 average = totalAges / numEntries;
162
163 //Display the new data
164 cout << "\n"
165     << "The oldest students age: " << highestAge << "( " <<
166     << "The youngest students age: " << lowestAge << "( " <<
167     << "Average age: " << average << endl;
168
169 break;
170
171 case '6': //Quits the Program
172     return 0;
173
174
175 default: //Input Validation
176     cout << "Invalid option. Please try again.\n";
177 }
178 }
179 }
180 }
181 }

```

```

PROBLEMS TERMINAL ... Code + - - - - -
garrettjackson@Garretts-MacBook-Pro Problem 2 % cd "/Users/garrettjackson/C++ Programming Class/Labs/Lab 06/Lab Problems/Problem 2/" && g++ menu-highandlow.cpp -o menu-highandlow && ./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 - Trinity
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 - Trinity
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 - Trinity
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( Trinity )
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
Options: 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";
}

}

}
```