

Name: Garrett Jackson

Structures

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 2. Weather Statistics

Write a program that uses a structure to store the following weather data for a particular month:

- Total Rainfall
- High Temperature
- Low Temperature
- Average Temperature

The program should have an array of 12 structures to hold weather data for an entire year. When the program runs, it should ask the user to enter data for each month. (The average temperature should be calculated.)

Once the data are entered for all the months, the program should calculate and display the average monthly rainfall, the total rainfall for the year, the highest and lowest temperatures for the year (and the months they occurred in) (use index), and the average of all the monthly average temperatures.

Input Validation: Only accept temperatures within the range between -100 and +140 degrees Fahrenheit.

USE THE NEXT TEMPLATE (MANDATORY)

```
//DO NOT MODIFY THIS SECTION
#include <iostream>
#include <fstream>
using namespace std;

struct Weather
{
    int rainfall;
    int hTemp;
    int lTemp;
    float avg;
};

//prototypes (USE ONLY IF YOU WILL USE FUNCTIONS)

int main()
{
    //USE THIS DATA ONLY FOR TESTS.
    //THE DATA MUST BE INTRODUCED BY THE USER OR FROM FILE
    Weather year[ 12 ] = { 200, 90, 60, 0.0,
                          300, 95, 62, 0.0,
```

```
                200,  99, 65, 0.0,  
                200, 101, 66, 0.0,  
                400, 105, 67, 0.0,  
                600, 108, 70, 0.0,  
                700, 112, 72, 0.0,  
                700, 111, 74, 0.0,  
                400, 108, 72, 0.0,  
                200, 104, 68, 0.0,  
                200,  98, 66, 0.0,  
                100,  92, 64, 0.0 };  
  
//ifstream ifile;  
int highest, lowest, totalRain=0, fullAverage;  
float sumAvg=0;  
//ADD YOUR CODE FROM HERE
```

Output (for this data):

```
Enter rainfall, highest and lowest temperature for month #1: 200 90 60  
Enter rainfall, highest and lowest temperature for month #2: 300 95 62  
Enter rainfall, highest and lowest temperature for month #3: 200 99 65  
Enter rainfall, highest and lowest temperature for month #4: 200 101 66  
Enter rainfall, highest and lowest temperature for month #5: 400 105 67  
Enter rainfall, highest and lowest temperature for month #6: 600 108 70  
Enter rainfall, highest and lowest temperature for month #7: 700 112 72  
Enter rainfall, highest and lowest temperature for month #8: 700 111 74  
Enter rainfall, highest and lowest temperature for month #9: 400 108 72  
Enter rainfall, highest and lowest temperature for month #10: 200 104 68  
Enter rainfall, highest and lowest temperature for month #11: 200 98 66  
Enter rainfall, highest and lowest temperature for month #12: 100 92 64  
Averages of the year:  
75  
78.5  
82  
83.5  
86  
89  
92  
92.5  
90  
86  
82  
78  
Total rainfall: 4200  
Highest temperature: 112 on month #7  
Lowest temperature: 60 on month #1  
Average temperature: 84.5417
```

```

//main.cpp Problem 1 Weather Statistics U X
main.cpp Problem 2 Multiple Sentences U
Problem 1 Weather Statistics > main.cpp > getData(Weather [])
28 int main()
29 {
30     //Function calls for each requirement
31     getData(year);
32     findAverage(year, totalRain, fullAverage, sumAvg, index);
33     highest = highestTemp(year, highIndex);
34     lowest = lowestTemp(year, lowIndex);
35
36     //Display a list of the values necessary
37     cout << "Total Rain: " << totalRain << "\n"
38           << "Highest Temp: " << highest << "Month is " << highIndex + 1 << " " << "\n"
39           << "Lowest Temp: " << lowest << "Month is " << lowIndex + 1 << " " << "\n"
40           << "Average Rainfall: " << fullAverage << "\n"
41           << "Average Temperature: " << sumAvg / 12 << endl;
42
43     return 0;
44 }
45
46 // Functions Start Here
47
48 void getData(Weather year[])
49 {
50     for(int i = 0; i < 12; i++)
51     {
52         //Displays month number
53         cout << "Enter data for month " << i + 1 << " : ";
54
55         //Rainfall
56         cout << "Total Monthly Rainfall, Highest Temp, and Lowest Temp: ";
57         cin >> year[i].rainfall >> year[i].hTemp >> year[i].lTemp;
58
59         // Input validation between -100 and 140
60         while(year[i].hTemp < -100 || year[i].hTemp > 140 ||
61               year[i].lTemp < -100 || year[i].lTemp > 140 ||
62               year[i].rainfall < 0 || year[i].rainfall > 100)
63             continue;
64     }
65 }
66
67 //PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
68
69 Monthly Average for month 3: 82
70 Monthly Average for month 4: 83.5
71 Monthly Average for month 5: 86
72 Monthly Average for month 6: 89
73 Monthly Average for month 7: 92
74 Monthly Average for month 8: 92.5
75 Monthly Average for month 9: 90
76 Monthly Average for month 10: 86
77 Monthly Average for month 11: 82
78 Monthly Average for month 12: 78
79
80 Total Rain: 4200
81 Highest Temp: 112(Month is 7)
82 Lowest Temp: 68(Month is 1)
83 Average Rainfall: 350
84 Average Temperature: 84.5417
85
86 garrettjackson@Garretts-MacBook-Pro Problem 1 Weather Statistics %

```

//DO NOT MODIFY THIS SECTION

```

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

```

```

struct Weather
{
    int rainfall;
    int hTemp;
    int lTemp;
    float avg;
};

```

//prototypes (USE ONLY IF YOU WILL USE FUNCTIONS)

```

void getData(Weather year[]);
int highestTemp(Weather year[], int &index);
int lowestTemp(Weather year[], int &index);
void findAverage(Weather year[], int &totalRain, int &fullAverage, float &sumAvg,
int &index);

```

```

int main()
{
    //USE THIS DATA ONLY FOR TESTS.
    //THE DATA MUST BE INTRODUCED BY THE USER OR FROM FILE
    /* Weather year[ 12 ] = {  200,  90, 60, 0.0,
                                300,  95, 62, 0.0,
                                200,  99, 65, 0.0,
                                200, 101, 66, 0.0,
                                400, 105, 67, 0.0,
                                600, 108, 70, 0.0,

```

```

        700, 112, 72, 0.0,
        700, 111, 74, 0.0,
        400, 108, 72, 0.0,
        200, 104, 68, 0.0,
        200, 98, 66, 0.0,
        100, 92, 64, 0.0 };

*/
//ifstream ifile;
int highest, lowest, totalRain=0, fullAverage;
float sumAvg=0;
//ADD YOUR CODE FROM HERE

int index, highIndex, lowIndex;

// Declare structure array
Weather year[ 12 ]; // 12 months in a year

//Function calls for each requirement
getData(year);

findAverage(year, totalRain, fullAverage, sumAvg, index);

highest = highestTemp(year, highIndex);
lowest = lowestTemp(year, lowIndex);

//Display a list of the values necessary
cout << "\nTotal Rain: " << totalRain << "\n"
    << "Highest Temp: " << highest << "(Month is " << highIndex + 1 << ")"
<< "\n"
    << "Lowest Temp: " << lowest << "(Month is " << lowIndex + 1 << ")" <<
"\n"
    << "Average Rainfall: " << fullAverage << "\n"
    << "Average Temperature: " << sumAvg / 12 << endl;

return 0;
}

// Functions Start Here

void getData(Weather year[])
{
    for(int i = 0; i < 12;i++)
    {
        //Displays month number
        cout << "Enter data for month " << i + 1 << ":\n";

        //Rainfall
        cout << "Total Monthly Rainfall, Highest Temp, and Lowest Temp: ";
        cin >> year[i].rainfall >> year[i].hTemp >> year[i].lTemp;

        // Input validation between -100 and 140
        while(year[i].hTemp < -100 || year[i].hTemp > 140 ||
            year[i].lTemp < -100 || year[i].lTemp > 140)
        {

```

```
        cout << "Invalid temperature. Please re-enter high and low temp for
month #" << i + 1 << ": ";
        cin >> year[i].hTemp >> year[i].lTemp;
    }
}

//Finds highest temperature
int highestTemp(Weather year[], int &index)
{
    int highest = year[0].hTemp;
    index = 0; // Used for Displaying the highest month number
    for(int i = 0; i < 12; i++)
    {
        if(year[i].hTemp > highest)
        {
            highest = year[i].hTemp;
            index = i;
        }
    }

    return highest;
}

//Find the lowest temperature
int lowestTemp(Weather year[], int &index)
{
    int lowest = year[0].lTemp;
    index = 0;
    for(int i = 0; i < 12; i++)
    {
        if(year[i].lTemp < lowest)
        {
            lowest = year[i].lTemp;
            index = i;
        }
    }
    return lowest;
}

//Averages monthly rainfall
void findAverage(Weather year[], int &totalRain, int &fullAverage, float &sumAvg,
int &index)
{
    cout << "Averages of the year:\n";
    for (int i = 0; i < 12; i++)
    {
        index = i;

        // Sum the total rainfall
        totalRain += year[i].rainfall;

        // Calculates the average of the temperatures for each month and stores
in the structure avg
        year[i].avg = (year[i].hTemp + year[i].lTemp) / 2.0f;
        cout << "Monthly Average for month" << " " << index + 1 << ": "<<
year[i].avg << "\n\n";

        //Sums the avg of each months averages
```

```
        sumAvg += year[i].avg;
    }
    // Average the total rainfall
    fullAverage = totalRain / 12;
}
```

Problem3. Multiple sentences and file

Modify the previous program so it will be able to read the data from a file or write to it if the user requests it. Use a menu, for:

- 1) Read from the file
- 2) Read from the keyboard
- 3) Display data including calculated information.
- 4) Write to file
- 5) Exit

Example of file "weather.txt":

```
200 90 60
300 95 62
200 99 65
200 101 66
400 105 67
600 108 70
700 112 72
700 111 74
400 108 72
200 104 68
200 98 66
100 92 64
```

```

20 int main()
21 {
22     while (true)
23     {
24         // 1: Display data, including the calculations\n
25         cout << "1: Display Data\n";
26         cout << "2: Write Data to the file\n";
27         cout << "3: Exit the program\n";
28         cin >> options;
29
30         // Set a switch case that allows user to choose options
31         switch (options)
32         {
33             // User chooses to read from a file
34             case 1:
35                 if (!ifstream("weather.txt"))
36                 {
37                     cout << "Error opening the file listed, try again.\n";
38                     return 1;
39                 }
40
41                 for (int i = 0; i < 12; i++)
42                 {
43                     if (!year[i].rainfall)
44                         year[i].rainfall = year[i].hTemp;
45                     year[i].lTemp = year[i].lTemp;
46                     year[i].avg;
47                 }
48                 if (!year[i].close());
49                 break;
50             // User chooses to input from the keyboard
51             case 2:
52                 getData(year);
53                 break;
54             // Display the data including the calculated information
55             case 3:
56                 findAverage(year, totalRain, fullAverage, sumAvg, index);
57                 highest = highestTemp(year, highIndex);
58                 lowest = lowestTemp(year, lowIndex);
59         }
60     }
61 }

```

Terminal Output:

```

Total Rain: 4200
Highest Temp: 112(Month is 7)
Lowest Temp: 68(Month is 1)
Average Rainfall: 350
Average Temperature: 84.5417
Select from the Menu
1: Read Data File
2: Enter Data Manually
3: Display Data, including the calculations
4: Write Data to the file
5: Exit the program
4
Data successfully written to weather.txt
Select from the Menu
1: Read Data File
2: Enter Data Manually
3: Display Data, including the calculations
4: Write Data to the file
5: Exit the program

```

// DO NOT MODIFY THIS SECTION

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
struct Weather
```

```
{
```

```
    int rainfall;
```

```
    int hTemp;
```

```
    int lTemp;
```

```
    float avg;
```

```
};
```

```
// prototypes (USE ONLY IF YOU WILL USE FUNCTIONS)
```

```
void getData(Weather year[]);
```

```
int highestTemp(Weather year[], int &index);
```

```
int lowestTemp(Weather year[], int &index);
```

```
void findAverage(Weather year[], int &totalRain, int &fullAverage, float &sumAvg, int &index);
```

```
int main()
```

```
{
```

```
    // USE THIS DATA ONLY FOR TESTS.
```

```
    // THE DATA MUST BE INTRODUCED BY THE USER OR FROM FILE
```

```
    /*
```

```
    Weather year[ 12 ] = { 200, 90, 60, 0.0,
```

```
                           300, 95, 62, 0.0,
```

```
                           200, 99, 65, 0.0,
```

```
                           200, 101, 66, 0.0,
```

```
                           400, 105, 67, 0.0,
```

```
                           600, 108, 70, 0.0,
```

```
                           700, 112, 72, 0.0,
```

```
                           700, 111, 74, 0.0,
```

```
                           400, 108, 72, 0.0,
```

```
                           200, 104, 68, 0.0,
```

```
                           200, 98, 66, 0.0,
```

```
                           100, 92, 64, 0.0 };
```



```
*/

ifstream ifile;

int highest, lowest, totalRain = 0, fullAverage;

float sumAvg = 0;


int highIndex, lowIndex;


// Declare structure array

Weather year[12]; // 12 months in a year

ofstream ofile;

int options, index;


while (true)
{
    cout << "Select from the Menu\n"
        << "1: Read Data File\n"
        << "2: Enter Data Manually\n"
        << "3: Display Data, including the calculations\n"
        << "4: Write Data to the file\n"
        << "5: Exit the program\n"
        << endl;

    cin >> options;


    // Setup a switch case that allows user to choose options
```

```
switch (options)
{
    // User chooses to read from a file
    case 1:
        iffile.open("weather.txt");

        if (!ifile)
        {
            cout << "Error opening the file listed, try again.";
            return 1;
        }

        for(int i = 0; i < 12; i++)
        {
            ifile >> year[i].rainfall
                >> year[i].hTemp
                >> year[i].lTemp
                >> year[i].avg;
        }

        ifile.close();

        break;

    // User chooses to input from the keyboard
    case 2:
        getData(year);
```

```
break;

// Display the data including the calculated information
case 3:

    findAverage(year, totalRain, fullAverage, sumAvg, index);

    highest = highestTemp(year, highIndex);
    lowest = lowestTemp(year, lowIndex);

    // Display a list of the values necessary
    cout << "\nTotal Rain: " << totalRain << "\n"
         << "Highest Temp: " << highest << "(Month is " << highIndex + 1 << ")" << "\n"
         << "Lowest Temp: " << lowest << "(Month is " << lowIndex + 1 << ")" << "\n"
         << "Average Rainfall: " << fullAverage << "\n"
         << "Average Temperature: " << sumAvg / 12 << endl;

    break;

// Write the information to the file
case 4:

    ofile.open("weather.txt");

    if (!ofile)
    {
```

```
        cout << "Error opening file for writing." << endl;

        break;
    }

    for (int i = 0; i < 12; i++)
    {
        ofile << year[i].rainfall << " "

            << year[i].hTemp << " "

            << year[i].lTemp << " "

            << year[i].avg << endl;
    }

    ofile.close();

    cout << "Data successfully written to weather.txt" << endl;

    break;

// Exit the program

case 5:

    cout << "Exiting program" << endl;

    return 1;

}

}

return 0;

}

// Functions Start Here
```

```
// Read user input data
```

```
void getData(Weather year[])
```

```
{
```

```
    for (int i = 0; i < 12; i++)
```

```
    {
```

```
        // Displays month number
```

```
        cout << "Enter data for month " << i + 1 << ":\n";
```

```
        // Rainfall
```

```
        cout << "Total Monthly Rainfall, Highest Temp, and Lowest Temp: ";
```

```
        cin >> year[i].rainfall >> year[i].hTemp >> year[i].lTemp;
```

```
        // Input validation between -100 and 140
```

```
        while (year[i].hTemp < -100 || year[i].hTemp > 140 ||
```

```
               year[i].lTemp < -100 || year[i].lTemp > 140)
```

```
        {
```

```
            cout << "Invalid temperature. Please re-enter high and low temp for month #" << i + 1 << ":  
";
```

```
            cin >> year[i].hTemp >> year[i].lTemp;
```

```
        }
```

```
    }
```

```
}
```

```
// Finds highest temperature
```

```
int highestTemp(Weather year[], int &index)
{
    int highest = year[0].hTemp;

    index = 0; // Used for Displaying the highest month number

    for (int i = 0; i < 12; i++)
    {
        if (year[i].hTemp > highest)
        {
            highest = year[i].hTemp;

            index = i;
        }
    }

    return highest;
}
```

// Find the lowest temperature

```
int lowestTemp(Weather year[], int &index)
{
    int lowest = year[0].lTemp;

    index = 0;

    for (int i = 0; i < 12; i++)
    {
        if (year[i].lTemp < lowest)
```

```
{
    lowest = year[i].lTemp;

    index = i;
}

}

return lowest;
}

// Averages monthly rainfall

void findAverage(Weather year[], int &totalRain, int &fullAverage, float &sumAvg, int &index)
{
    cout << "Averages of the year:\n";

    for (int i = 0; i < 12; i++)
    {
        index = i;

        // Sum the total rainfall

        totalRain += year[i].rainfall;

        // Calculates the average of the temperatures for each month and stores in the structure avg

        year[i].avg = (year[i].hTemp + year[i].lTemp) / 2.0f;

        cout << "Monthly Average for month" << " " << index + 1 << ": " << year[i].avg << "\n\n";

        //Sums the avg of each months averages
```

```
    sumAvg += year[i].avg;  
}  
  
// Average the total rainfall  
fullAverage = totalRain / 12;  
}
```