

1. Rainfall Statistics

Write a program that lets the user enter the total rainfall for each of the 12 months into an array of double. The program should calculate and display the total rainfall for the year, the average monthly rainfall, and the months with the highest and lowest amounts (month numbers).

Input Validation: Do not accept negative numbers for monthly rainfall figures.

Source Code:

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

//prototypes (USE ONLY IF YOU WILL USE FUNCTIONS)
void read(double[]);
double total(double[]);
void highLow(double[], int&, int&);

int main()
{
    //initialization only for tests, commenting the input part
    double rainfall[12]; //={2.3, 5.6, 8.2, 15.2, 4.7, 3.7, 6.8, 3.1, 1.1, 6.6, 2.7,
5.5 };
    int max, min;
    //ADD YOUR CODE FROM HERE
    //1. Ask user to input data
    read(rainfall);

    //2. Print the results based on the entered data
    cout << "Total: " << total(rainfall) << endl;
    cout << "Average: " << total(rainfall) / 12 << endl;
    highLow(rainfall, max, min);
    cout << "Highest month: " << max << endl;
    cout << "Lowest month: " << min << endl;
    return 0;
}

//3. Define functions
//3.1 Write the read function
void read(double arr[])
{
    for (int i = 0; i < 12; i++)
    {
        //3.1.1 Ask for the rainfall for each month and store it in an array
        //3.1.2 Use a do while loop for input validation
        do {
            cout << "Rainfall month " << i + 1 << ": ";
            cin >> arr[i];
            if (arr[i] < 0)
                cout << "Please do not enter a negative number for the rainfall for
this month." << endl;
        } while (arr[i] < 0);
    }
}
```

```

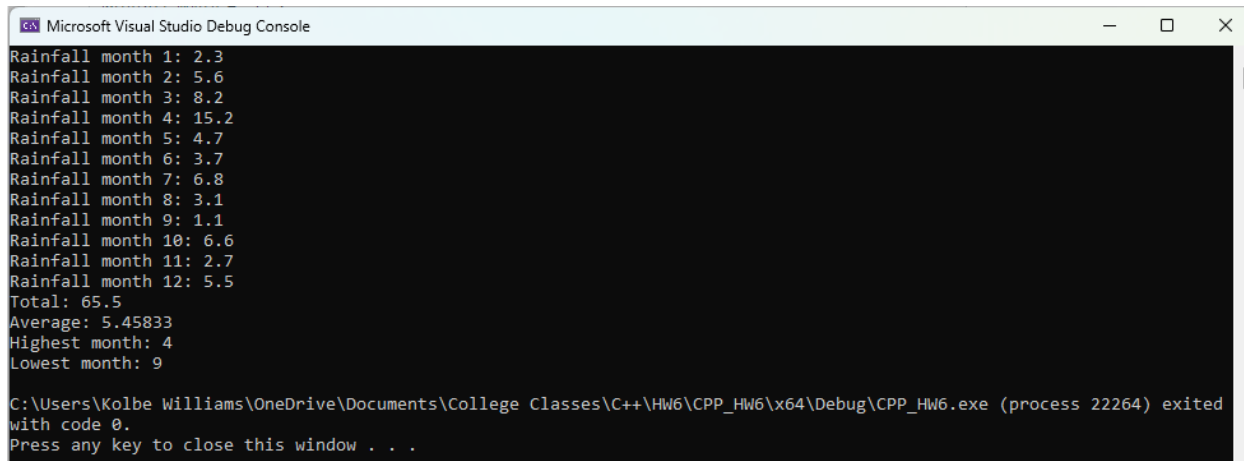
}

//3.2 Write the total function
double total(double arr[])
{
    //3.2.1 Add all of the elements of the array together to get the total
    double sum = 0.0;
    for (int i = 0; i < 12; i++)
        sum += arr[i];
    return sum;
}

//3.3 Write the highLow function
void highLow(double arr[], int& high, int& low)
{
    //3.3.1 Initialize variables for the high and low values
    double highValue = arr[0];
    double lowValue = arr[0];
    for (int i = 0; i < 11; i++)
    {
        //3.3.2 Go through each element of the array and reassign the highest value
        //if necessary
        if (highValue < arr[i + 1])
        {
            highValue = arr[i + 1];
            //3.3.3 identify the number of the highest month and assign it to the
            //reference variable
            high = i + 2;
        }
        //3.3.4 Repeat the steps used to find the mighest rainfall month for the
        //lowest rainfall month
        if (lowValue > arr[i + 1])
        {
            lowValue = arr[i + 1];
            low = i + 2;
        }
    }
}
}

```

Output:



```

Microsoft Visual Studio Debug Console
Rainfall month 1: 2.3
Rainfall month 2: 5.6
Rainfall month 3: 8.2
Rainfall month 4: 15.2
Rainfall month 5: 4.7
Rainfall month 6: 3.7
Rainfall month 7: 6.8
Rainfall month 8: 3.1
Rainfall month 9: 1.1
Rainfall month 10: 6.6
Rainfall month 11: 2.7
Rainfall month 12: 5.5
Total: 65.5
Average: 5.45833
Highest month: 4
Lowest month: 9

C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\HW6\CPP_HW6\x64\Debug\CPP_HW6.exe (process 22264) exited
with code 0.
Press any key to close this window . . .

```

```
Microsoft Visual Studio Debug Console
Rainfall month 1: 2.3
Rainfall month 2: 5.6
Rainfall month 3: 8.2
Rainfall month 4: 15.2
Rainfall month 5: -4.7
Please do not enter a negative number for the rainfall for this month.
Rainfall month 5: 4.7
Rainfall month 6: 3.7
Rainfall month 7: 6.8
Rainfall month 8: 3.1
Rainfall month 9: 1.1
Rainfall month 10: 6.6
Rainfall month 11: 2.7
Rainfall month 12: 5.5
Total: 65.5
Average: 5.45833
Highest month: 4
Lowest month: 9

C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\HW6\CPP_HW6\Debug\CPP_HW6.exe (process 26368) exited
with code 0.
Press any key to close this window . . .
```

2.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

Source Code:

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

//prototypes (INSERT HERE IF YOU USE FUNCTIONS)
void lowestHighest(int[], int, int&, int&);
int totalNum(int[], int);
//end prototypes
int main()
{
    ifstream ifile;
    string fileName;
    int lowest, highest, total = 0, counter = 0, numbers[20];
    double average;
    cout << "Name of file: ";
    cin >> fileName;
    ifile.open(fileName);
    if (ifile.fail())
    {
        cout << "Error" << endl;
```

```

        return 1;
    }
    //ADD YOUR CODE FROM HERE
    //1. Read numbers from file, put them into the array, and determine the length
of the array
    while (!ifile.eof())
    {
        ifile >> numbers[counter];
        counter++;
    }
    ifile.close();

    //2. Calculate the required values using function calls
    lowestHighest(numbers, counter, lowest, highest);
    total = totalNum(numbers, counter);
    average = total / static_cast<double>(counter);

    //3. Display output
    cout << "Lowest number: " << lowest << endl;
    cout << "Highest number: " << highest << endl;
    cout << "Total: " << total << endl;
    cout << "Average: " << average << endl << endl;
    cout << "File \"" << fileName << "\" for the example:" << endl << endl;
    for (int i = 0; i < counter; i++)
        cout << numbers[i] << endl;
    return 0;
}

//4. Write the lowestHighest function
void lowestHighest(int arr[], int length, int& low, int& high)
{
    //4.1 Pass variables by reference to hold the highest and lowest values
    low = high = arr[0];
    //4.2 Traverse through the array and compare each element to find the highest
and lowest element
    for (int i = 0; i < length - 1; i++)
    {
        if (low > arr[i + 1])
            low = arr[i + 1];
        if (high < arr[i + 1])
            high = arr[i + 1];
    }
}

//5. Write the totalNum function
int totalNum(int arr[], int length)
{
    int sum = 0;
    //5.1 Traverse through the array and add all of the elements together to get the
total
    for (int i = 0; i < length; i++)
        sum += arr[i];
    return sum;
}

```

Output:

