```
Script started on 2024-05-05 15:23:30-05:00 [TERM="xterm" TTY="/dev/pts/0" COLUMNS=
ee43254@ares:~$ pwd
/home/students/ee43254
ee43254@ares:~$ cat statnumbs.info
Name: Kyle Enkhzul

Class: CSC122-W01



Activity: Round 'em Up!

Level: 1.5, 1.5 (base program)



Description:



This lab provides practice with file processing where we don't know the amount

of information in the file ahead of time. It calculates basic statistics for a

set of numbers stored in a file. ee43254@ares:~$ show-code statnumbs.cpp


statnumbs.cpp:
```

```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <cmath>
#include <limits>

// Function to calculate the average of a vector of numbers
double calculateAverage(const std::vector<double>& numbers) {
    double sum = 0.0;
    for (double num : numbers) {
        sum += num;
    }
    return sum / static_cast<double>(numbers.size()); // Explicit cast to
}

// Function to calculate the variance of a vector of numbers
double calculateVariance(const std::vector<double>& numbers, double average
    double variance = 0.0;
    for (double num : numbers) {
        variance += (num - average) * (num - average);
    }
    return variance / static_cast<double>(numbers.size() - 1); // Explicit
}

// Function to calculate the standard deviation from variance
double calculateStandardDeviation(double variance) {
    return sqrt(variance);
}

int main() {

        using namespace std;

    string filename;
    cout << "Enter the name of the file: ";
    getline(cin, filename);

    ifstream file(filename);
    if (!file) {
        cerr << "Error: Could not open the file.\n";
        return 1;
    }

    vector<double> numbers;
    double number;
    while (file >> number) {
        numbers.push_back(number);
    }

    if (numbers.empty()) {
        cerr << "Error: No numbers found in the file.\n";
        return 1;
    }

    double average = calculateAverage(numbers);
    double variance = calculateVariance(numbers, average);
    double standardDeviation = calculateStandardDeviation(variance);

        // set max and min to -infinity and infinity for reusability
    double max = -numeric_limits<double>::infinity();
    double min = numeric_limits<double>::infinity();
    for (double num : numbers) {
        if (num > max) {
            max = num;
        }
        if (num < min) {
            min = num;
        }
    }

        cout << "Count: " << numbers.size() << endl;
        cout << "Minimum: " << min << endl;
    cout << "Average: " << average << endl;
        cout << "Maximum: " << max << endl;
    cout << "Standard Deviation: " << standardDeviation << endl;



    return 0;
}
ee43254@ares:~$ CPP statnumbs
statnumbs.cpp***


ee43254@ares:~$ cat stats1.txt
 ee43254@ares:~$ cat stats2.txt
1 2 3 4

5 6 7 8



9 10ee43254@ares:~$ cat stats3.txt
97 32 54 68 23 89 12 45 78 50

10 82 64 39 18 91 77 21 46 3

29 61 72 6 95 16 83 38 4 69

57 8 17 86 25 71 96 31 52 42

14 98 19 75 27 93 2 60 85 47

63 80 35 7 49 26 65 94 59 81

70 20 33 56 9 53 11 88 36 76

1 90 55 24 67 41 99 37 74 51

22 66 30 84 13 58 5 87 48 62

40 79 43 15 100 44 28 73 34 92ee43254@ares:~$ ./statnumbs.out
Enter the name of the file: stats1.txt
Error: No numbers found in the file.
ee43254@ares:~$ ./statnumbs.out
Enter the name of the file: stats2.txt
Count: 10
Minimum: 1
Average: 5.5
Maximum: 10
Standard Deviation: 3.02765
ee43254@ares:~$ ./statnumbs.out
Enter the name of the file: stats3.txt
Count: 100
Minimum: 1
Average: 50.5
Maximum: 100
Standard Deviation: 29.0115
ee43254@ares:~$ cat statnumbs.tpq
1. Does spacing between the numbers matter?

No, the spacing between the numbers does not matter. The program reads

numbers from the file regardless of the spacing between them.



2. Do they have to be in order (ascending/descending)?

No, the numbers do not have to be in any particular order (ascending

or descending). The program processes the numbers as they appear in the file.



3. Do they have to be integers? Floating point?

The program is designed to handle both integers and floating-point numbers.

It reads numbers from the file and stores them as doubles, which can represent

 both integers and floating-point numbers.



4. Is it possible for your program to run out of space while reading the file?

No, it's not possible for the program to run out of space while reading the

file. The program reads the file line by line and processes one number at a

time. It doesn't store all the numbers in memory at once, so memory usage is

minimal and won't cause issues even for large files.



5. When finding the largest/smallest item in a list, what value should you

start with as your assumed smallest/largest value?

When finding the largest item in a list, you should start with the smallest

possible value that the data type can represent, and when finding the

smallest item, you should start with the largest possible value that the

data type can represent. In the case of doubles, you can start with positive

and negative infinity as the initial values for max and min, respectively,

because any valid number will be greater than negative infinity and

less than positive infinity.ee43254@ares:~$ exit
exit
```

```
Script done on 2024-05-05 15:24:20-05:00 [COMMAND_EXIT_CODE="0"]
```