Barthelemy Peter

bapeter

CS1300 <AB>

Program 2

Functions.cpp

Due Date: 03/06/17

CS1300 Programming Project 3 Spring 2017

Functions and Arrays Mansfield, J

Due Friday March 3, 2017

Objective:

Design and implement a program which will read in a text file that will fill an array of 100 random integers.

Use the array of integers and appropriate functions to:

1. Print the array out in five columns (on screen and to a file)
2. Find the largest value – print with labels
3. Find the smallest value – print with labels
4. Print the array out in five columns starting at the highest index and going to the smallest index (reverse from a)
5. Calculate the mean, standard deviation, variance of the set of values
6. Print the answers calculated in step e with appropriate labels

Follow all of the directions for this program. Make sure to include all of your documentation for: a) planning; b) test and evaluation design; c) reflection.

The output from the program must be well organized and labeled. The output should go to both an output file and the screen.

Turn in all required documentation:

• Cover page

• Copy of assignment

• Design documents

• Source code

• Output listing

• Reflection

Additionally, drop tarred and gzipped file in cs1300 dropbox.

Due Date: 3/3/2017

Design Documents

Design and implement a program which will read in a text file that will fill an array of 100 random integers. This array must then be sorted from smallest to largest, and vice versa. The maximum and minimum as well as standard deviation, variance and average of the array should then be calculated. The absolute output of this information should be replicated in a text file simultaneously.

**Pre-conditions:**

* No user input
* All data comes from text file
* Mostly functions containing for loops will be used

**Post-conditions:**

* Average, variance, standard dev., min and max will be calculated
* All output should replicate to a text file
* Array data is manipulated, for loops, and functions are manipulated to produce output
* Data is clear, nicely formatted and visible to users

Source Code

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cmath>

using namespace std;

const int rows = 20;

const int columns = 5;

const int SIZE = 100;

float array[SIZE];

ifstream inFile;

ofstream outFile;

void input();

void increasing\_Array();

float maxValue();

float minValue();

void decreasing\_Array();

float sum(float array[], int num);

float variance(float average);

char space = ' ';

int main()

{

inFile.open("Array.txt");

outFile.open("arrayout.txt");

input();

float largest = maxValue();

float smallest = minValue();

float total = sum(std::array, SIZE);

float average = total / SIZE;

float var = variance(average);

float standard\_Dev = sqrt(var);

cout << endl;

increasing\_Array();

for (int i= 0; i < rows; i++)

{

for (int j= 0; j < columns; j++)

{

cout << left << setw(10) << array[i\*5+j] << space;

outFile << left << setw(10) << array[i\*5+j] << space;

}

cout << endl;

}

cout << endl << endl;

cout << "The largest value in the list of numbers is: " << largest << endl;

outFile << "The largest value in the list of numbers is: " << largest << endl;

cout << "The smallest value in the list of numbers is: " << smallest << endl << endl;

outFile << "The smallest value in the list of numbers is: " << smallest << endl << endl;

for (int i= 20; i > 0 ; i--)

int k = 0;

for (int j= 5; j > 0 ; j--)

{

++k;

cout << left << setw(10) << array[i\*5-k] << space ;

outFile << left << setw(10) << array[i\*5-1-k] << space ;

}

cout << endl;

}

cout << endl << endl;

cout << "This is the average of the array " << average << endl;

outFile << "This is the average of the array " << average << endl;

cout << "This is the variance of the array " << var << endl;

outFile << "This is the variance of the array " << var << endl;

cout << "This is the standard deviation of the array " << standard\_Dev << endl << endl;

outFile << "This is the standard deviation of the array " << standard\_Dev << endl << endl;

inFile.close();

outFile.close();

return 0;

}

void decreasing\_Array()

{

int temp;

int smallestIndex;

int location;

int i;

for (i = 0; i < SIZE - 1 ; i++) // these are the numbers for input

{

smallestIndex = i;

for (location = i + 1; location < SIZE; location++)

if (array[smallestIndex] > array[location])

smallestIndex = location;

temp = array[smallestIndex];

array[smallestIndex] = array[i];

array[i] = temp;

}

}

float maxValue()

{

float MAX;

for (int i = 0; i < SIZE; i++)

{

if (array[i] > array[i+1])

MAX = array[i];

}

return MAX;

}

float minValue()

{

float MIN;

for (int i = 1; i < SIZE ; i++)

{

if (array[i] < array[i-1])

MIN = array[i];

}

return MIN;

}

void increasing\_Array()

{

int temp;

int smallestIndex;

int location;

int i;

for (i = 0; i < SIZE - 1 ; i++) // these are the numbers for input

{

smallestIndex = i;

for (location = i + 1; location < SIZE; location++)

if (array[smallestIndex] > array[location])

smallestIndex = location;

temp = array[smallestIndex];

array[smallestIndex] = array[i];

array[i] = temp;

}

}

void input()

{

for (int i = 0; i < SIZE; i++)

inFile >> array[i];

}

float sum(float array[], int num)

{

float answer = 0;

for (int i = 0 ; i < SIZE; i++)

{

answer += array[i];

}

return answer;

}

float variance(float average)

{

float temp, ans, root;

float var= 0;

for (int i = 0; i < SIZE; i++)

{

temp = pow((array[i] - average), 2);

var += temp;

}

ans = var /(SIZE - 1);

return ans;

}

Output Listing

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

26 27 28 29 30

31 32 33 34 35

36 37 38 39 40

41 42 43 44 45

46 47 48 49 50

51 52 53 54 55

56 57 58 59 60

61 62 63 64 65

66 67 68 69 70

71 72 73 74 75

76 77 78 79 80

81 82 83 84 85

86 87 88 89 90

91 92 93 94 95

96 97 98 99 100

The largest value in the list of numbers is: 100

The smallest value in the list of numbers is: 1

100 99 98 97 96

95 94 93 92 91

90 89 88 87 86

85 84 83 82 81

80 79 78 77 76

75 74 73 72 71

70 69 68 67 66

65 64 63 62 61

60 59 58 57 56

55 54 53 52 51

50 49 48 47 46

45 44 43 42 41

40 39 38 37 36

35 34 33 32 31

30 29 28 27 26

25 24 23 22 21

20 19 18 17 16

15 14 13 12 11

10 9 8 7 6

5 4 3 2 1

This is the average of the array 50.5

This is the variance of the array 841.667

This is the standard deviation of the array 29.0115

Reflection

**Overview:**

The purpose of this program was to manipulate value returning and void functions to complete various tasks by manipulating an array its data. User input was unnecessary as it was assumed that all necessary input was imported from an existing file.

**Challenges and Solution:**

My biggest challenges with this program was trying to manipulate loops around the array, creating a loop that would output the array in descending order, having the loop output to a file and getting my code to compile. Often for loops would not work because I did not have a well enough understanding of the concept to receive the output requested. Most of my errors were resolved through observation, patience, and research. The biggest problem I had is still not resolved because my loop never compiled in Putty.

**Lessons Learned:**

I have learned that functions make the life of a programmer much easier since they eliminate the need for lengthy, redundant coding.