**LAB 8**

**SECTION G**

**SUBMITTED BY:**

**RYAN SAMUELSON**

**SUBMISSION DATE:**

**11/12/15**

# Problem

The program needed to calculate the running average of all of the items in the array and then move the buffer by taking off the first item and moving all the others left one and then put the new one on the end of the length. The program also needed to find the maximum and the minimum of all the data. The program also needed to end when the left button was pressed.

# Analysis

In analyzing the program I needed to fill the arrays and then only add one value from the end of the array. from the graphs, I can conclude that the only difference from the longer length from the shorter length is the averages are more consistent and less far apart.

# Design

The program had to be in two parts, one to fill the array and one to add the end of the existing array. For the first part, filling the array, I need a simple for loop. The program can’t run the functions with a non-filled array.

Second part of the program I needed to add a new number from the scanf line and add it to the end of the existing array. This was not so hard because all I needed to do in the function was place the previous item from “I + 1” into “I”. With the new array I was able to calculate the max and min and the averages.

# Testing

In my testing, the only problem I came across was the minimums and maximums. Before I changed the max and min pointers so they weren’t set so anything, I had them set to 0, the problem came when I ran the program. If the min didn’t go below 0, the min would always be set as 0. The same can go for the max as well.

Comments

**Lab8**

#include <stdio.h>

#define MAXPOINTS 10000

// compute the average of the first num\_items of buffer

double avg(double buffer[], int num\_items){

double total = 0;

double averageOfTotal;

int i;

for(i = 0; i < num\_items; i++){

total = buffer[i] + total;

}

averageOfTotal = total / num\_items;

return averageOfTotal;

}

//update the max and min of the first num\_items of array

void maxmin(double array[], int num\_items, double\* max, double\* min){

int i;

for(i = 0; i < num\_items; i++){

if (i == 0) {

\*max = array[i];

\*min = array[i];

}

if(array[i] > \*max){

\*max = array[i];

}

else if(array[i] < \*min){

\*min = array[i];

}

}

}

//shift length-1 elements of the buffer to the left and put the

//new\_item on the right.

void updatebuffer(double buffer[], int length, double new\_item){

int i;

int j = 0;

for(i = 0; i < length; i++){

buffer[i] = buffer[i + 1];

j++;

}

buffer[j] = new\_item;

}

int main(int argc, char\* argv[]) {

/\* DO NOT CHANGE THIS PART OF THE CODE \*/

double x[MAXPOINTS], y[MAXPOINTS], z[MAXPOINTS];

int lengthofavg = 0;

if (argc>1) {

sscanf(argv[1], "%d", &lengthofavg );

printf("You entered a buffer length of %d\n", lengthofavg);

}

else {

printf("Enter a length on the command line\n");

return -1;

}

if (lengthofavg <1 || lengthofavg >MAXPOINTS) {

printf("Invalid length\n");

return -1;

}

/\* PUT YOUR CODE HERE \*/

int t, b1, b2, b3, b4, b5, s;

double ax, ay, az;

double xmax, xmin, ymax, ymin, zmax, zmin;

int i;

for(i = 0; i < lengthofavg; i++){

scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3, &b4, &b5, &s );

if (b3 == 1) {

return 0;

}

x[i] = ax;

y[i] = ay;

z[i] = az;

}

while (b3 == 0) {

// do some stuff here

maxmin(x, lengthofavg, &xmax, &xmin);

maxmin(y, lengthofavg, &ymax, &ymin);

maxmin(z, lengthofavg, &zmax, &zmin);

printf("%lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf\n", ax, ay, az, avg(x, lengthofavg), avg(y, lengthofavg), avg(z, lengthofavg), xmax, xmin, ymax, ymin, zmax, zmin);

scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3, &b4, &b5, &s );

updatebuffer(x, lengthofavg, ax);

updatebuffer(y, lengthofavg, ay);

updatebuffer(z, lengthofavg, az);

}

}



