**Smoothing**

**LAB 8**

**SECTION 2**

**Gavin Monroe**

**SUBMISSION DATE:**

**11/07/2017**

# Problem

Smooth the axis of x, y, and z after a inputted number from the user. I solved this by using the example given. It took me three tries but I got it figured out by talking to a fellow classmate.

# Analysis

We need use arrays to hold data, and then replace and move the array around to make sure we can calculate the new average for each axis. We use 3 different inputs for 3 different axis’s.

# Design

We use three different functions to solve, calculate and solve the problem that we face. Then too find the max and min I use a simplistic way to find that by using a loop with a set already max and min. For the update buffer I used of some code from a classmate but modified it.

# Testing

I had to redo my code three different times until I got it right. And for the demo I didn’t have enough time, and when I went to go demo it on Friday all the controllers were gone so I couldn’t demo it. I got the code done shortly after class though. I had to solve and redo my code three different times because the lab instructions weren’t written that well.

# Comments

Please be more clear in the instructions

# Source Code

// SE 185 Lab 8

#include <stdio.h>

#define MAXPOINTS 10000

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

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

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

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

//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 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 on = 1;

double gx, gy, gz;

double \_maxX, \_maxY, \_maxZ;

double \_minX, \_minY, \_minZ;

do{

scanf("%lf, %lf, %lf", &gx, &gy, &gz);

updatebuffer(x, lengthofavg, gx);

updatebuffer(y, lengthofavg, gy);

updatebuffer(z, lengthofavg, gz);

maxmin(x, lengthofavg, &\_maxX, &\_minX);

maxmin(y, lengthofavg, &\_maxY, &\_minY); maxmin(z, lengthofavg, &\_maxZ, &\_minZ);

printf("%lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf\n", gx, gy, gz, avg(x, lengthofavg), avg(y, lengthofavg), avg(z, lengthofavg), \_maxX, \_minX, \_maxY, \_minY, \_maxZ, \_minZ);

}while(on == 1);

}

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

int i;

int i2;

double total = 0;

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

total += buffer[i];

}

return (total / num\_items);

}

void maxmin(double array[], int num\_items, double\* \_max, double\* \_min){

int i;

\*\_max=-99000;

\*\_min=99000;

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

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

\*\_max=array[i];

}

if (i != 0){

if (array[i]< \*\_min){

\*\_min=array[i];

}

}

}

}

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

int i;

for(i=1; i<length;i++){

buffer[i-1] = buffer[i];

}

buffer[length - 1] = new\_item;

}

# Screen Shots



