//Nirdesh Bhandari

// README lab 1

Hypothesis - For most cases there is a "correct order" with respect to overall performance to visit each location of large arrays.

Development:

For this lab I learned C programing from scratch with the help of youtube tutorials and online forums.

I learned about functions, variables, declaration, specifiers and pointers in great detail.

My submission includes my "Draft"s of code where I show how I progressed and polished my code.

Draft 1 - Very beginning, still struggling with a lot of things.

Draft 2 - Added measure of time, everything is still in the main function

Draft 3 - Added functions, cleaned things up a little.

Draft 4 - Organized code, removed Debugging print statements, indented everything.

Draft final - Separated functions, final polish.

My code runs and displays Raw and Normalized time for each platform.

It averages three runs for each x by x array and displays the time taken.

It also measures time for each individual traversal, sums them and averages them to show the difference.

I used the #lscpu command to get the BOGOMIPS value for each platform and set ulimit to 8mb on each.

Conclusions:

Row Major traversal seems to be much faster in C programming because row memory seems to be laid out linearly and could be accessed and worked on by C much faster.

Also I noticed that when I had everything in the Main code, which traversal came first made a difference.

I concluded that this could be because of the difference in working memory vs cache memory.

When measuring time individually for each individual traversal, the time turned out to be much more than when time was measured collectively. This I concluded was because the overall number of operations was much higher.

For RAW time measurements

For Normalized time measurements

Strech Goals:

1. My code does this and prints it in the top exactly like the output shown in the question paper.

B. When array is global rather than local, the time measurements will not be conclusive to show which traversal is better because of the difference in access times between working memory and cache memory.

The same results would apply to three-dimensional arrays but the difference would be much higher. This is because memory is laid out linearly in C and therefore traversal times depend on the type of traversal.