README.txt

Group Names:

1. Omid Azodi
2. Andrew Vu

File Names:

1. Matrix.h
2. Matrix.cpp
3. horizontal\_parallel.cpp
4. Parallel.cpp
5. vertical\_parallel.cpp
6. Serial.cpp
7. CreateMatrix.cpp
8. DataMatrixA.txt (These are the input files to read in data for MatrixA)
9. DataMatrixB.txt(These are the input files to read in data for MatrixB)

We have a make file called makefile. To open makefile just do emacs makefile. To use the makefile just type make and [enter] in the command line.

After typing make and [enter] in the command line, go ahead and press ls.

There should be 4 executables that are ready to run!

Names of 4 executables:

tables:

* Cmat.out
* horizontal\_parallel.out
* Serial.out
* vertical\_parallel.out
* Parallel.out

First run the Cmat by typing ./Cmat.out [enter]

Then you can test whichever type of multiplication you would like to have

1. Serial (type ./Serial.out [enter])
2. Parallel MxN (This has a max limit due to school server not allowing allocation of resources to make these approximately 4 million threads for the [2000 x 2000] [2000 x 2000] case, however the multiplication works completely correctly if you choose the correct bounds that it can hold. We tested our code to see what the exact number is for the thread limit and we found a limit at 302 threads were possible on the school servers. (type ./Parallel.out [enter])
3. Horizontal (This runs both normal core case and twice core count) (type ./ horizontal\_parallel.out [enter])
4. Vertical (This runs both normal core case and twice core count) (type ./ vertical\_parallel.out [enter])

We also have a PDF that has our performance pictured out for ALL 3 cases.

Test Cases:

1. A[2000 x 2000] \* B[2000 x 2000]
2. A[2000 x 200] \* B[200 x 2000]
3. A[1000x 10000] \* B[10000x 1000]

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

For the Pipes.cpp

Just compile like this:

g++ Pipes.cpp [enter]

./a.out [enter]

Then enter the message you want to encrypt Such as:

Hello World

And return should be the encrypted message:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_