Devin Hardy

00076619

[Dhardy2@my.athens.edu](mailto:Dhardy2@my.athens.edu)

ASG3

//The beginning of my Linux code

// Devin Hardy

#include <iostream>

#include <thread>

#include <vector>

#include <ctime>

#include <time.h>

// An attempt at a function that makes a nXn matrix

// it returns a pointer of an array of size\*size

// given the int size

int\* makeMatrix(int size)

{

int \*mtrx = new int[size\*size];

srand(time(0));

for(int i=0; i< size\*size; i++)

{

\*(mtrx + i) = (rand() % 100);

}

return mtrx;

}

// function that multiplies matrixes

// imperically

int\*\* multiMatrix(int\* mtrx, int size)

{

int \*mtrx2 = makeMatrix(size);

int\*\* mMtrx = new int\* [size];

int \*result = new int [size];

int count = 1;

for(int n=0, N = size\*size; n < N; ++n)

{

int i = n/ size;

int j = n% size;

float temp = 0;

for (int k = 0; k<size; ++k)

{

temp += mtrx[i\*N+k] \* mtrx2[k\*N+j];

}

result[i]= temp;

if(count !=0 && count%size == 0)

{

count=0;

}

else

count++;

}

}

int main()

{

int mtrxSize1 = 100;

int mtrxSize2 = 500;

int count = 1;

int\* mtrx1 = makeMatrix(mtrxSize1);

int\* mtrx2 = makeMatrix(mtrxSize2);

std::cout << mtrxSize1 << " matrix" << std::endl;

for(int i=0; i < (mtrxSize1\*mtrxSize1); i++)

{

std::cout << \*mtrx1 << " ";

mtrx1++;

if(count % mtrxSize1 == 0)

{

std::cout << std::endl;

count++;

}

else

count++;

}

std::cout << std::endl << std::endl << std::endl;

return 0;

}

Cited

matrix - Multiplying two matrices using pointers in C++. (n.d.). Retrieved February 8, 2023, from Stack Overflow website: https://stackoverflow.com/questions/61461878/multiplying-two-matrices-using-pointers-in-c

‌