//Author: Will Irvin

//Homework Week 3: 2D Array Stuff Part Deux

//

//This program will perform calculations with 2D arrays that can handle

//any array up to 100 by 100. There will also be functions that will be

//responsible to do one process only.

//

//I used the following resources to aid in building and troubleshooting this program…

//www.stackoverflow.com, www.cplusplus.com, and fellow students.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

//This is the maximum for this 2D Array.

const int arrySIZE = 100;

//Function Prototypes.

void addROWS(int ary[][arrySIZE], int, int, int, int),

addCOLS(int ary[][arrySIZE], int, int, int, int),

subtrtROWS(int ary[][arrySIZE], int, int, int, int),

subtrtCOLS(int ary[][arrySIZE], int, int, int, int),

replace(int ary[][arrySIZE], int, int, int),

print(int ary[][arrySIZE], int, int, ofstream &),

copyROWS(int ary[][arrySIZE], int, int, int, int);

int rowVALUES(int ary[][arrySIZE], int, int, int),

colVALUES(int ary[][arrySIZE], int, int, int),

get(int ary[][arrySIZE], int, int);

int main()

{

ofstream outFile;

int firstARRY[arrySIZE][arrySIZE] = { { 0 } },

secondARRY[arrySIZE][arrySIZE] = { { 0 } },

maxROW, maxCOL, lgRowVAL, smColVAL,

cValTEMP;

outFile.open("2DArrayStuffOutPartDeux.txt");

//Display info for the output file functionResults.

outFile << "Author: Will Irvin" << endl;

outFile << "Student ID: 00061459" << endl;

outFile << "CS318: Mondays 2:30 - 5:20" << endl;

outFile << "Homework Week 3: 2D Array Stuff Part Deux\n" << endl;

//The User supplys the dimensions of the array.

cout << "How many Rows would you like to enter? ";

cin >> maxROW;

cout << "How many Columns would you like to enter? ";

cin >> maxCOL;

outFile << "\n\n\ Part Deux Array Results" << endl;

outFile << "\n\n The Results for the New Array!" << endl;

outFile << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

///// THE CALLS /////

//The setup of the new array with 7 rows and 6 columns.

int i = 1;

for (int r = 0; r < maxROW; r++)

{

for (int c = 0; c < maxCOL; c++)

{

replace(firstARRY, r, c, i);

}

i++;

}

//Multiply rows 0 thru 6 by 5.

int j = 1;

for (int cCOUNT = 0; cCOUNT < maxCOL; cCOUNT++)

{

for (int rCOUNT = 0; rCOUNT < maxROW; rCOUNT++)

{

replace(firstARRY, rCOUNT, cCOUNT, j \* 5);

}

j++;

}

print(firstARRY, maxROW, maxCOL, outFile);

outFile.close();

system("pause");

return 0;

}

///// THE FUNCTIONS /////

////////////////////////////////////////////////////////////////////

// This is function adds any two rows //

////////////////////////////////////////////////////////////////////

void addROWS(int ary[][arrySIZE], int r, int c, int ROW1, int ROW2)

{

for (int cCOUNT = 0; cCOUNT < c; cCOUNT++)

{

ary[ROW2][cCOUNT] = ary[ROW2][cCOUNT] + ary[ROW1][cCOUNT];

}

}

/////////////////////////////////////////////////////////////////////

// This function adds any two columns //

/////////////////////////////////////////////////////////////////////

void addCOLS(int ary[][arrySIZE], int r, int c, int COL1, int COL2)

{

for (int rCOUNT = 0; rCOUNT < r; rCOUNT++)

{

ary[rCOUNT][COL1] = ary[rCOUNT][COL1] + ary[rCOUNT][COL2];

}

}

////////////////////////////////////////////////////////////////////////////////////

// This function subtracts any two rows //

////////////////////////////////////////////////////////////////////////////////////

void subtrtROWS(int ary[][arrySIZE], int r, int c, int rSUBTRACT1, int rSUBTRACT2)

{

for (int cCOUNT = 0; cCOUNT < c; cCOUNT++)

{

ary[rSUBTRACT2][cCOUNT] = ary[rSUBTRACT2][cCOUNT] - ary[rSUBTRACT1][cCOUNT];

}

}

////////////////////////////////////////////////////////////////////////////////////

// This function subtracts any two columns //

////////////////////////////////////////////////////////////////////////////////////

void subtrtCOLS(int ary[][arrySIZE], int r, int c, int cSUBTRACT1, int cSUBTRACT2)

{

for (int cCOUNT = 0; cCOUNT < r; cCOUNT++)

{

ary[cCOUNT][cSUBTRACT2] = ary[cCOUNT][cSUBTRACT2] - ary[cCOUNT][cSUBTRACT1];

}

}

///////////////////////////////////////////////////////////////

// This function finds the largest row value //

///////////////////////////////////////////////////////////////

int rowVALUES(int ary[][arrySIZE], int r, int c, int rSEARCH)

{

int rTEMP = ary[rSEARCH][0];

for (int cCOUNT = 0; cCOUNT < c; cCOUNT++)

{

if (rTEMP < ary[rSEARCH][cCOUNT])

{

rTEMP = ary[rSEARCH][cCOUNT];

}

}

return rTEMP;

}

//////////////////////////////////////////////////////////////

// This function finds the largest column value //

//////////////////////////////////////////////////////////////

int colVALUES(int ary[][arrySIZE], int r, int c, int cSEARCH)

{

int cTEMP = ary[0][cSEARCH];

for (int rCOUNT = 0; rCOUNT < r; rCOUNT++)

{

if (cTEMP > ary[rCOUNT][cSEARCH])

{

cTEMP = ary[rCOUNT][cSEARCH];

}

}

return cTEMP;

}

//////////////////////////////////////////

// This is the get function //

//////////////////////////////////////////

int get(int ary[][arrySIZE], int r, int c)

{

int temp;

temp = ary[r][c];

return temp;

}

/////////////////////////////////////////////////////////////

// This is the replace function //

/////////////////////////////////////////////////////////////

void replace(int ary[][arrySIZE], int r, int c, int newVAL)

{

//This will replace one value for another in the array.

ary[r][c] = newVAL;

}

///////////////////////////////////////////////////////////////////

// This is the copy function //

///////////////////////////////////////////////////////////////////

void copyROWS(int ary[][arrySIZE], int r, int c, int ROW1, int ROW2)

{

for (int cCOUNT = 0; cCOUNT < c; cCOUNT++)

{

ary[ROW2][cCOUNT] = ary[ROW1][cCOUNT];

}

}

/////////////////////////////////////////////////////////////////

// This is the print function //

/////////////////////////////////////////////////////////////////

void print(int ary[][arrySIZE], int r, int c, ofstream &outFile)

{

for (int rCOUNT = 0; rCOUNT < r; rCOUNT++)

{

for (int cCOUNT = 0; cCOUNT < c; cCOUNT++)

{

outFile << setw(5) << ary[rCOUNT][cCOUNT];

}

outFile << endl;

}

return;

}