//Objective: Create Magic Square 2

//Name: Ethan Chiu

//libraries

#include <fstream> //for file input

#include <iostream> //for input and output

#include <string> //for using string built in functions

using namespace std;

void makeMagicSquare (int \*\*magicSquare, int dimension);

int computeMagicNumber (int \*\*magicSquare, int dimension);

bool validateInput (int \*\*magicSquare, int number, int a, int b, int dimension);

bool determineIfMagic (int \*\*magicSquare, int dimension, int magicNumber);

int main()

{

//Data

int dimension;

int magicNumber;

//Enter dimension for magic square

cout << "Enter dimension for magic square: ";

cin >> dimension;

cin.ignore(1000, 10);

int\*\* magicSquare = new int\*[dimension];

for(int i = 0; i < dimension; ++i)

{

magicSquare[i] = new int[dimension];

}

//magicSquare = new int[dimension][dimension];

makeMagicSquare(magicSquare, dimension);

magicNumber = computeMagicNumber (magicSquare, dimension);

if(determineIfMagic(magicSquare, dimension, magicNumber))

{

cout << "This is a magic square" << endl;

}

else

{

cout << "This is NOT a magic square" << endl;

}

}

void makeMagicSquare (int \*\*magicSquare, int dimension)

{

bool invalidNumber = false;

//Enter data for magic square

cout << "Enter numbers of magic square (left to right, top to bottom): " << endl;

//Add to array

for(int i = 0; i < dimension; i++)

{

for(int b = 0; b < dimension; b++)

{

do

{

cout << "Row " << i << ", Column " << b << ": ";

cin >> magicSquare[i][b];

cin.ignore(1000,10);

} while (!validateInput(magicSquare, magicSquare[i][b], i, b, dimension));

}

}

}

bool validateInput (int \*\*magicSquare, int number, int a, int b, int dimension)

{

//Validate input

if(number > dimension\*dimension || number < 1)

{

cout << "Not in range" << endl;

return false;

}

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

{

for (int d = 0; d <= a; d++)

{

if(number == magicSquare[d][c])

{

cout << "Not unique" << endl;

return false;

}

}

}

return true;

}

int computeMagicNumber (int \*\*magicSquare, int dimension)

{

int sum = 0;

//Add to sum

for(int i = 0; i < dimension; i++)

{

for(int b = 0; b < dimension; b++)

{

sum += magicSquare[i][b];

}

}

return sum/dimension;

}

bool determineIfMagic (int \*\*magicSquare, int dimension, int magicNumber)

{

int sumColumn = 0;

int sumRow = 0;

int diagonal1 = 0;

int diagonal2 = 0;

//Check if it's a magic square (columns, rows)

for(int i = 0; i < dimension; i++)

{

sumRow = 0;

sumColumn = 0;

for(int b = 0; b < dimension; b++)

{

sumColumn += magicSquare[i][b];

sumRow += magicSquare[b][i];

}

if(sumRow != magicNumber || sumColumn != magicNumber)

{

return false;

}

}

//Check if diagonal is the same. Fix the diagonals

for(int i = 0; i < dimension; i++)

{

for(int b = 0; b < dimension; b++)

{

if (i == b)

{

diagonal1 += magicSquare[i][b];

}

}

}

for(int i = dimension - 1; i >= 0; i--)

{

for(int b = dimension - 1; b >= 0; b--)

{

if (i == (dimension - 1) - b )

{

diagonal2 += magicSquare[i][b];

}

}

}

//Last check

if(magicNumber != diagonal1 || magicNumber != diagonal2)

{

return false;

}

else

{

return true;

}

}