// T.p tin Array2D.h

#include <iostream>

#include <vector>

using namespace std;

#ifndef \_my\_Array2D

#define \_my\_Array2D

#define defaultSZ 10

// Template class definition for 2.dimension array

template <class T>

class array2D{

typedef vector<T> arrayT;

vector<arrayT> data;

arrayT dummy;

public:

array2D(int m=0, int n=0){

if(m<0) m=-m;

if(n<0) n=-n;

if(m==0) m=defaultSZ;

if(n==0) n=defaultSZ;

dummy.resize(n);

data.resize(m);

for(int i=0; i<data.size(); i++)

data[i].resize(n);

}

arrayT& operator[](int i){

if(i<0 || i>=data.size()){

cout << "Subscript out of range!" << endl;

return dummy;

}

return data[i];

}

int nRow(){

return data.size();

}

int nCol(){

return data[0].size();

}

friend operator >> (istream& inDev, array2D<T>& m);

friend operator << (ostream& outDev, array2D<T>& m);

};

template <class T>

operator >> (istream& inDev, array2D<T>& matrix){

const int m=matrix.nRow(), n=matrix.nCol();

for(int i=0; i<m; i++){

cout<<"Row "<<i<<":"<<endl;

for(int j=0; j<n; j++){

cout<<" a["<<i<<"]["<<j<<"]=";

cin>>matrix[i][j];

}

}

}

template <class T>

operator << (ostream& outDev, array2D<T>& matrix){

const int m=matrix.nRow(), n=matrix.nCol();

for(int i=0; i<m; i++){

for(int j=0; j<n; j++){

cout<<matrix[i][j]<<" ";

}

cout<<endl;

}

}

#endif