/\*\*

\* Tìm đúng ma trận nghịch đảo bằng phương pháp Gauss- jordan

\*\*/

#include<iostream>

#include <fstream>

using namespace std;

// đọc ma trận

void read(int &n,float matrix[100][100])

{

ifstream f("D:/duyen/gaussjd.txt" );

f>> n;

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

{

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

f>>matrix[i][j];

}

f.close();

// sau khi tìm đc ma trận nghịch đảo, ghi ma trận vào file

}

void write(int n,float matrix[100][100])

{

ofstream f;

f.open("D:/duyen/gaussjd.txt", ios::app );

f<<"\n\n\n";

f<<"ma tran nghich dao la\n";

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

{

for(int j=n;j<2\*n;j++)

{

f<<matrix[i][j];

f<<" \t ";}

f<<"\n";

}

f.close();

}

int main()

{

int i = 0, j = 0, k = 0, n = 0;

float d = 0.0;

float matrix[100][100];

read(n,matrix);

cout << "Ma tran ban dau: " << endl;

for (i = 0; i < n; ++i)

{

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

{

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

}

cout << endl;

}

cout << endl;

// Lập ma trận mở rộng

for(i = 0; i < n; ++i)

{

for(j = 0; j < 2\*n; ++j)

{

if(j == (i + n))

{

matrix[i][j] = 1;

}

}

}

// Sắp xếp lại các hàng trong ma trận mở rộng

for(i = n; i > 1; --i)

{

if( matrix[i-1][1] < matrix[i][1] )

{

for(int j = 0; j < 2\*n; ++j)

{

swap(matrix[i][j] , matrix[i-1][j] );

}

}

}

//giảm thành ma trận đường chéo

for(i = 0; i < n; ++i)

{

for(j = 0; j < 2\*n; ++j)

{

if(j != i)

{

d = matrix[j][i] / matrix[i][i];

for(k = 0; k < 2\*n; ++k)

{

matrix[j][k] = matrix[j][k] - d \* matrix[i][k];

}

}

}

}

cout << endl;

//Giảm thành ma trận đơn vị

for(i = 0; i < n; ++i)

{

d = matrix[i][i];

for(j = 0; j < 2\*n; ++j)

{

matrix[i][j] = matrix[i][j] / d;

}

}

// in ra ma trận nghịch đảo

cout<<"Ma tran nghich dao:" << endl;

for(i=0; i < n; ++i)

{

for(j = n; j < 2\*n; ++j)

{

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

}

cout << endl;

}

write(n,matrix);

}