**HỌ VÀ TÊN: HUỲNH HỮU HOÀNG**

**LỚP: 22T\_DT1**

**MSSV: 102220063**

#include<bits/stdc++.h>

using namespace std;

#define MAX 100

// check doi xung

bool checkSymmetric(double a[100][100], int n){

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

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

if(i != j){

if(a[i][j] != a[j][i]) return false;

}

}

}

return true;

}

//mang con

void subMatrix(double a[100][100], int n, int x, int y, double submatrix[100][100]){

// double \*\*submatrix = new double\*[n - 1];

// double submatrix[MAX][MAX];

int subi = 0;

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

// submatrix[subi] = new double[n - 1];

int subj = 0;

if(i == y) continue;

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

if(j == x) continue;

submatrix[subi][subj] = a[i][j];

subj++;

}

subi++;

}

// return submatrix;

}

//tinh det

double determinant(double a[100][100], int n){

int det = 0;

if(n == 2){

return a[0][0] \* a[1][1] - a[1][0] \* a[0][1];

}

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

double submatrix[MAX][MAX];

subMatrix(a, n, x, 0, submatrix);

det += ((x % 2 == 0 ? 1 : - 1) \* a[0][x] \* determinant(submatrix, n - 1));

}

return det;

}

//check ma tran xac dinh duong

bool checkDeterminationPositive(double a[100][100], int n){

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

double submatrix[MAX][MAX];

subMatrix(a, n, i, i, submatrix);

double det = determinant(submatrix, i + 1);

cout << det << endl;

if(det <= 0) return false;

}

return true;

}

//cholesky co dien

void CholeskiDecompositionClassic(double a[100][100], int n){

int lower[100][100];

memset(lower, 0, sizeof(lower));

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

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

double sum = 0;

if(i == j){

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

sum += pow(lower[j][k], 2);

}

lower[j][j] = sqrt(a[j][j] - sum);

}else{

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

sum += (lower[i][k] \* lower[j][k]);

}

lower[i][j] = (a[i][j] - sum) / lower[j][j];

}

}

}

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

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

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

}

cout << "\t";

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

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

}

cout << endl;

}

}

// cholesky bienthe

void CholeskiDecompositionVariant(double a[100][100], int n){

double lower[100][100];

memset(lower, 0, sizeof(lower));

for(int i = 0 ; i < n ; i++) lower[i][i] = 1;

double d[100];

memset(d, 0, sizeof(d));

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

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

double sum = 0;

if(i == j){

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

sum += (pow(lower[j][k], 2) \* d[k]);

}

d[j] = a[j][j] - sum;

}else{

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

sum += (lower[i][k] \* lower[j][k] \* d[k]);

}

lower[i][j] = (a[i][j] - sum) / d[j];

}

}

}

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

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

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

}

cout << "\t\t\t";

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

if(i == j) cout << d[j] << " " ;

else cout << 0 << " ";

}

cout << "\t\t\t";

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

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

}

cout << endl;

}

}

int main(){

int n; cin >> n;

double a[n][100];

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

for(int j = 0 ; j < n ; j++) cin >> a[i][j];

}

cout << determinant(a, n) << endl;

CholeskiDecompositionClassic(a, n);

if(checkDeterminationPositive(a, n)){

}else{

cout << "Ma tran khong xac dinh duong\n";

}

CholeskiDecompositionVariant(a, n);

}

