**Матрицы (задание)**

## Задача 1

Напишите программу, которая вычисляет сумму элементов матрицы.

Программа должна вывести одно число – сумму элементов матрицы.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {}, sum = 0;

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

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

M[i][j] = 1 + rand() % 3;

cout << setw(3) << M[i][j];

sum = sum + M[i][j];

}

cout << endl;

}

cout << endl << sum;

return 0;

## }

## Задача 2

Напишите программу, которая определяет, сколько раз встречается в матрице элемент, равный *K*.

Программа должна вывести количество элементов матрицы, равных *K*.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {}, K, count = 0;

cin >> K;

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

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

cin >> M[i][j];

if (M[i][j] == K)

count++;

}

cout << endl;

}

cout << endl << count;

return 0;

}

### Задача 3

Напишите программу, которая находит минимальный и максимальный элементы в матрице. Если в матрице есть несколько одинаковых минимальных (максимальных) элементов, нужно найти индексы первого такого элемента в порядке обхода по строкам: сверху вниз, слева направо.

**Выходные данные**

В первой строке программа должна вывести индексы минимального элемента (строку и столбец), а затем – его значение. Нумерация строк и столбцов начинается с единицы. Все числа разделены пробелами. Во второй строке выводится информация о максимальном элементе в том же формате.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {};

int l\_max = 0, c\_max = 0, max = 0;

int l\_min = 0, c\_min = 0, min = 0;

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

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

cin >> M[i][j];

}

}

max = min = M[0][0];

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

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

if (M[i][j] > max) {

max = M[i][j];

l\_max = i;

c\_max = j;

}

if (M[i][j] < min) {

min = M[i][j];

l\_min = i;

c\_min = j;

}

}

}

cout << endl;

cout << l\_max + 1 << " " << c\_max + 1 << " " << max << endl;

cout << l\_min + 1 << " " << c\_min + 1 << " " << min << endl;

return 0;

}

## Задача 4

Напишите программу, которая находит в матрице строку с минимальной суммой.

Программа должна вывести все элементы найденной строки с минимальной суммой, разделив их пробелами. Если строк с одинаковой минимальной суммой несколько, нужно выбрать из них строку с минимальным индексом.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {};

int sum = 0;

int min = 0, line\_min = 0;

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

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

cin >> M[i][j];

}

}

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

min += M[0][j];

}

for (int i = 1; i < N; i++) {

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

sum += M[i][j];

}

if (sum < min) {

min = sum;

line\_min = i;

}

}

cout << endl;

cout << M[line\_min][0] << " " << M[line\_min][1] << " " << M[line\_min][2];

return 0;

}

## Задача 5

Напишите программу, которая находит в матрице столбцы, в которых есть элемент, равный максимальному.

**Выходные данные**

Программа должна вывести все столбцы, в которых есть элемент, равный максимальному элементу в матрице. Каждый столбец выводится в одну строку, элементы разделяются пробелами.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {};

int max = 0;

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

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

cin >> M[i][j];

}

}

max = M[0][0];

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

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

if (M[i][j] > max) {

max = M[i][j];

}

}

}

cout << endl;

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

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

if (M[i][j] == max) {

cout << M[0][j] << " " << M[1][j] << " " << M[2][j] << " " << endl;

break;

}

}

}

return 0;

}

## Задача 6

Напишите программу, которая обнуляет все элементы квадратной матрицы, расположенные выше главной диагонали.

**Выходные данные**

Программа должна вывести матрицу, у которой все элементы выше главной диагонали нулевые, а остальные элементы такие же, как в исходной матрице.

#include <iostream>

#include <random>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

srand(time(0));

const int N = 3;

int M[N][N] = {};

int max = 0;

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

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

cin >> M[i][j];

}

}

cout << endl;

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

for (int j = i + 1; j < N; j++) {

M[i][j] = 0;

}

}

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

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

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

}

cout << endl;

}

return 0;

}