Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа № 14

По дисциплине «Основы алгоритмизации и программирования»

На тему «Многомерные массивы»

Выполнил:

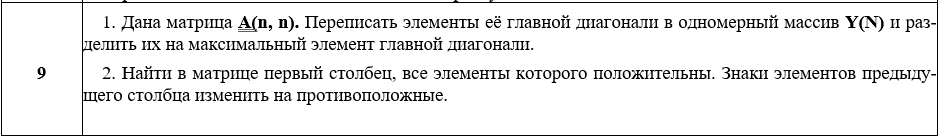
Студент 1 курса 7 группы

Ленкевич Павел Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант 9.**

****

**Задание 1.**

#include <iostream> // Подключение библиотек

#include <locale>

using namespace std; // Позволяет не писать std перед операторами

int main() { // Тело программы

setlocale(LC\_ALL, "ru"); // Подключаем русский язык

srand(time(NULL)); // Для создания случайных чисел

const int n = 5; // Подключаем переменные

double A[n][n], Y[n], max = 0; // Подключение многомерного массива

for (int i = 0; i < n; i++) { // Заполняем массив случайными числами

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

A[i][d] = (int)(((double)rand() / (double)RAND\_MAX) \* 100);

}

}

for (int i = 0; i < n; i++) { // Перемещаем элементы главной диагонали матрицы в новый массив

Y[i] = A[i][i];

if (A[i][i] > max) max = A[i][i]; // Определяет максимальный элемент массива

}

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

Y[i] = Y[i] / max;

}

for (int i = 0; i < n; i++) { // Заполняет второй массив числами

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

cout << A[i][d] << " ";

}

cout << endl;

}

cout << endl << "Наибольший элемент: " << max; // Вывод значений переменных

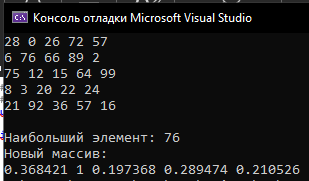
cout << endl << "Новый массив:\n";

for (int i = 0; i < n; i++) { // Вывод получившегося массива

cout << Y[i] << " ";

}

}

****

**Задание 2.**

#include <iostream> // Подключение библиотек

#include <locale>

using namespace std; // Позвояет не писать std перед каждым оператором

int main() {

setlocale(LC\_ALL, "ru");// Подключает русский язык

srand(time(NULL)); // Даёт возможность записи случайных чисел

const int n = 5;

int A[n][n], column; // Многомерный массив

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

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

A[i][d] = (int)(((double)rand() / (double)RAND\_MAX) \* 200 - 50); // Заполняем массив случайными числами

}

}

for (int i = 0; i < n; i++) { // Выводит элементы массива на экран

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

cout << A[i][d] << " ";

}

cout << endl;

}

bool done = false; // Логическая переменная

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

for (int d = 0; d < n; d++) { // Меняет знаки отрицательного столбика на противоположные, если удовлетворяет условиям

if (A[d][i] < 0) {

done = true;

break;

}

if (d == (n - 1)) {

if (i > 0) {

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

A[k][i - 1] = -A[k][i - 1];

}

}

column = i;

done = false;

}

}

if (done == false) { // Выводит все положительные колонки

cout << "Неотрицательные колонны\n";

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

cout << A[d][column] << endl;

}

cout << endl;

if (i > 0) { // Если условие выполняется, выводит элементы массива

cout << "Колонна до нее с противоположным знаком\n";

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

cout << A[d][column - 1] << endl;

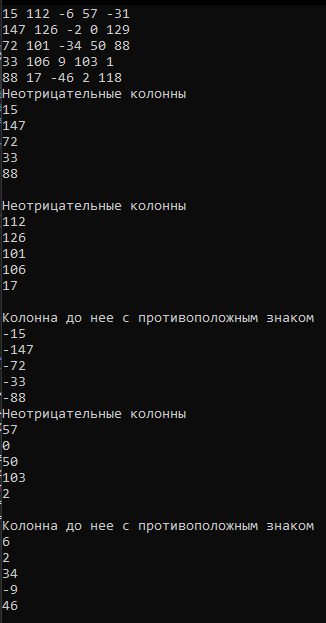
}

}

}

}

}

****

**Дополнительные задания**

**Задание 1.**

Дана квадратная матрица порядка **2n**, элементы которой формируются случайным образом и находятся в пределах от −10 до 10. Получить новую матрицу, переставляя ее блоки размера **n×n** в соответствии со схемой:

#include <iostream> // Подключение библиотек

#include <locale>

using namespace std; // Позволяет не писать std перед каждым оператором

int main() {

setlocale(LC\_ALL, "ru"); // Подключает русский язык

srand(time(NULL)); // Даёт возможность рандомизировать числа

const int n = 4;

int A[2 \* n][2 \* n], column; // Объявление многомерного массива

for (int i = 0; i < (2 \* n); i++) { // Заполнение массива случайными числами

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

A[i][d] = (int)(((double)rand() / (double)RAND\_MAX) \* 20 - 10);

}

}

for (int i = 0; i < (2 \* n); i++) { // Вывод элементов массива на экран

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

cout << A[i][d] << " ";

}

cout << endl;

}

for (int i = 0; i < (2 \* n); i++) { // Повторное заполнение массива числами, изменяемыми согласно схеме в лабораторной

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

int change = A[d][i];

A[d][i] = A[2 \* n - 1 - d][2 \* n - 1 - i];

A[2 \* n - 1 - d][2 \* n - 1 - i] = change;

}

}

cout << endl;

cout << "Результат массива: \n"; // Вывод элементов массива после преобразования

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

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

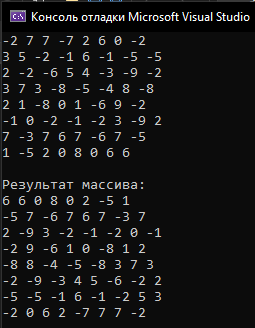
cout << A[i][d] << " ";

}

cout << endl;

}

}

****

**Задание 2.**

Латинским квадратом порядка **n** называется квадратная таблица размером **nхn**, каждая строка и каждый столбец которой содержат все числа от 1 до **n**. Для заданного **n** в матрице **L(n, n**) построить латинский квадрат порядка **n**.

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором

int main() {

const int n = 7;

int A[n][n], number = 1; // Объявляем многомерный массив

for (int i = 0; i < n; i++) { // Заполняем массив согласно условию

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

A[i][d] = number;

number++;

if (number > n) number = 1;

}

number++;

}

for (int i = 0; i < n; i++) { // Выводим элементы массива на экран

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

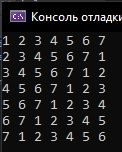
cout << A[i][d] << " ";

}

cout << endl;

}

}

****

**Задание 3.**

Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине − в позиции (2, 2), следующий − в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ.

#include <iostream> // Подключаем библиотеки

#include <locale>

using namespace std; // Позволяет не писать std перед каждым оператором

int main() {

srand(time(NULL)); // Даёт возможность рандомизировать числа

setlocale(LC\_ALL, "rus"); // Подключает русский язык

const int n = 5;

int A[n][n], changeid[2], checked = 0, max = 0; // Объявляем многомерный массив

for (int i = 0; i < n; i++) { // Заполняем массив случайными числами

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

A[i][d] = (int)(((double)rand() / (double)RAND\_MAX) \* 100);

}

}

for (int i = 0; i < n; i++) { // Выводим элементы массива на экран

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

cout << A[i][d] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++) { // Проверяем какой максимальный элемент массива

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

if (A[i][d] > max) {

max = A[i][d];

changeid[0] = i;

changeid[1] = d;

}

}

}

int change = A[checked][checked];

A[checked][checked] = max; // Подключаем дополнительные массивы

A[changeid[0]][changeid[1]] = change;

checked++;

for (int i = 1; i < n; i++) { // Перемещаем элементы массива пока максимальный элемент не окажется в позиции 1х1

int newmax = 0;

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

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

if (A[d][k] > newmax && A[d][k] < max) {

newmax = A[d][k];

changeid[0] = d;

changeid[1] = k;

}

}

}

change = A[checked][checked];

A[checked][checked] = newmax;

A[changeid[0]][changeid[1]] = change;

checked++;

max = newmax;

}

cout << endl << "Новый массив с максимальными значениями по диагонали:\n";

for (int i = 0; i < n; i++) { // Вывод элементов нового массива

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

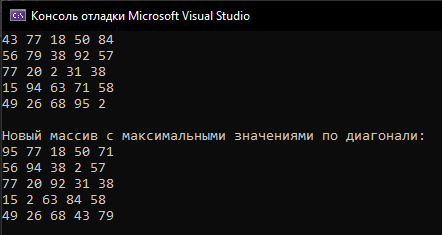
cout << A[i][d] << " ";

}

cout << endl;

}

}

****