Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 14

По дисциплине «Основы алгоритмизации и программирования»

на тему «**Многомерные массивы**»

Выполнила:

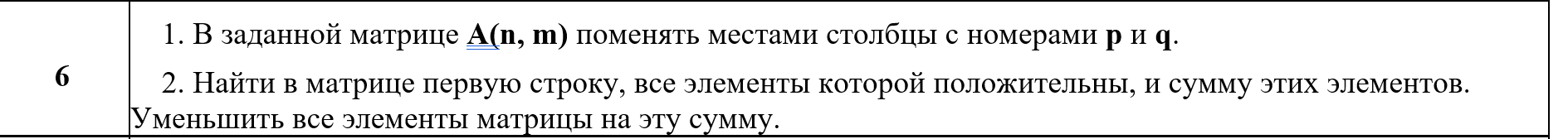
Студент(ка) 1 курса 7 группы

Гриценко Анна Александровна

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 6



Задание 1

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "RUS");

int n, m;//инициализация переменных

// Ввод размеров матрицы

cout << "Введите количество строк: ";

cin >> n;

cout << "Введите количество столбцов: ";

cin >> m;

// Выделение памяти под матрицу

int matrix[100][100];

// Ввод элементов матрицы

cout << "Введите элементы матрицы:" <<endl;

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

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

cout << "A[" << i << "][" << j << "]: ";

cin >> matrix[i][j];

}

}

int p, q;//инициализация переменных

// Ввод номеров столбцов для обмена

cout << "Введите номер первого столбца (p): ";

cin >> p;

cout << "Введите номер второго столбца (q): ";

cin >> q;

if (p >= 0 && p < m && q >= 0 && q < m) {

// Обмен значений столбцов p и q в каждой строке

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

int temp = matrix[i][p];

matrix[i][p] = matrix[i][q];

matrix[i][q] = temp;

}

// Вывод измененной матрицы

cout << "Матрица после обмена столбцов:" <<endl;

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

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

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

}

cout << endl;

}

}

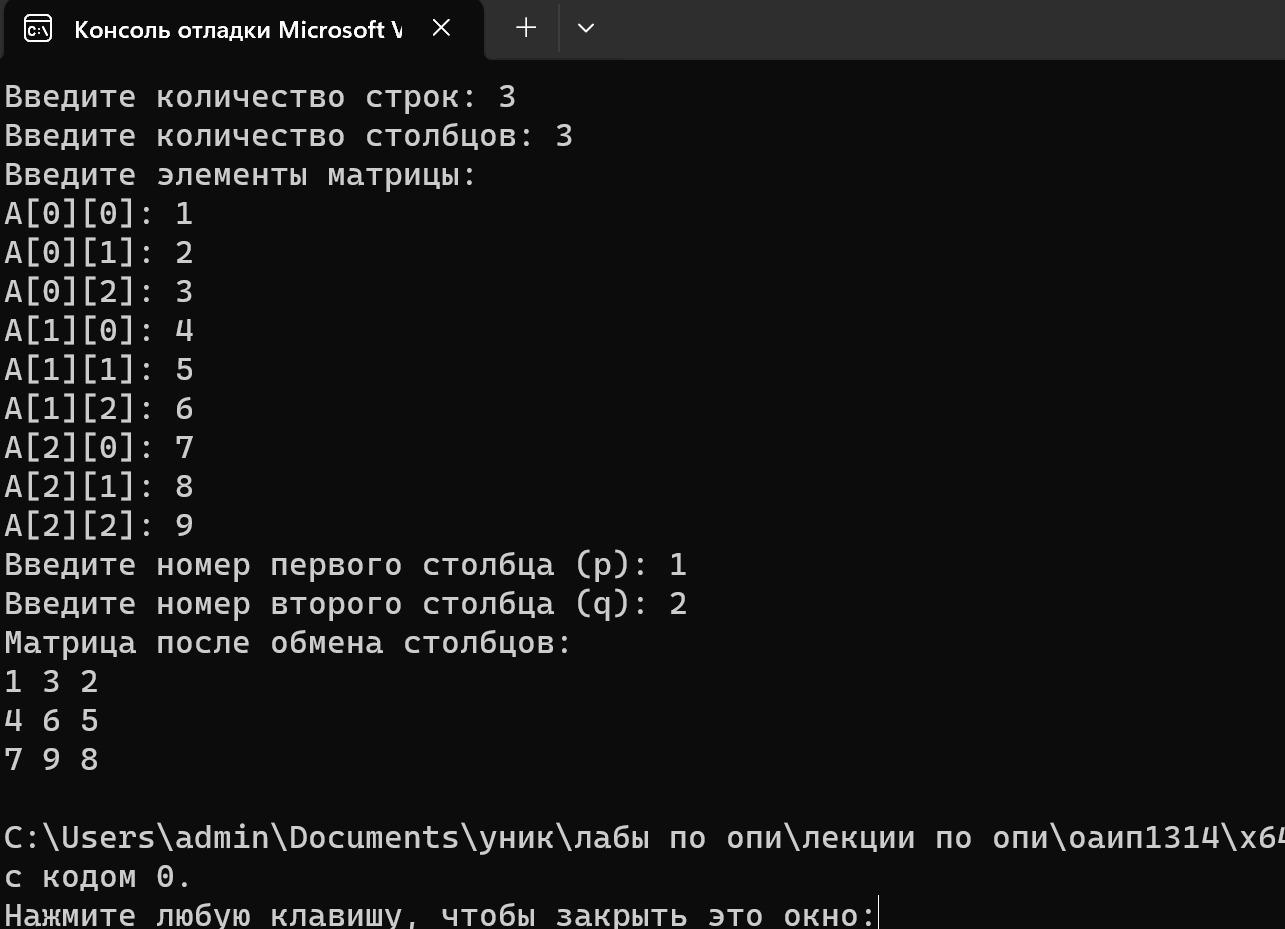
else {

cout << "Ошибка: Некорректные номера столбцов." << endl;

}

return 0;

}



Задание 2

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "RUS");

int n, m;//инициализация переменных

// Ввод размеров матрицы

cout << "Введите количество строк: ";

cin >> n;

cout << "Введите количество столбцов: ";

cin >> m;

// Выделение памяти под матрицу

int\* matrix = new int[n \* m];

// Ввод элементов матрицы

cout << "Введите элементы матрицы:" << endl;

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

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

cout << "A[" << i << "][" << j << "]: ";

cin >> \*(matrix + i \* m + j);

}

}

int\* positiveRow = nullptr;

int sumOfPositiveRow = 0;

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

bool allPositive = true;

// Проверка, все ли элементы строки положительны

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

if (\*(matrix + i \* m + j) <= 0) {

allPositive = false;

break;

}

}

if (allPositive) {

positiveRow = matrix + i \* m;

// Нахождение суммы элементов положительной строки

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

sumOfPositiveRow += \*(positiveRow + j);

}

break;

}

}

if (positiveRow != nullptr) {

// Уменьшение всех элементов матрицы на сумму положительной строки

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

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

\*(matrix + i \* m + j) -= sumOfPositiveRow;

}

}

cout << "Найдена первая положительная строка. Сумма элементов: " << sumOfPositiveRow << endl;

// Вывод измененной матрицы

cout << "Матрица после уменьшения элементов:" << endl;

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

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

cout << \*(matrix + i \* m + j) << " ";

}

cout <<endl;

}

}

else {

cout << "В матрице нет положительных строк." << endl;

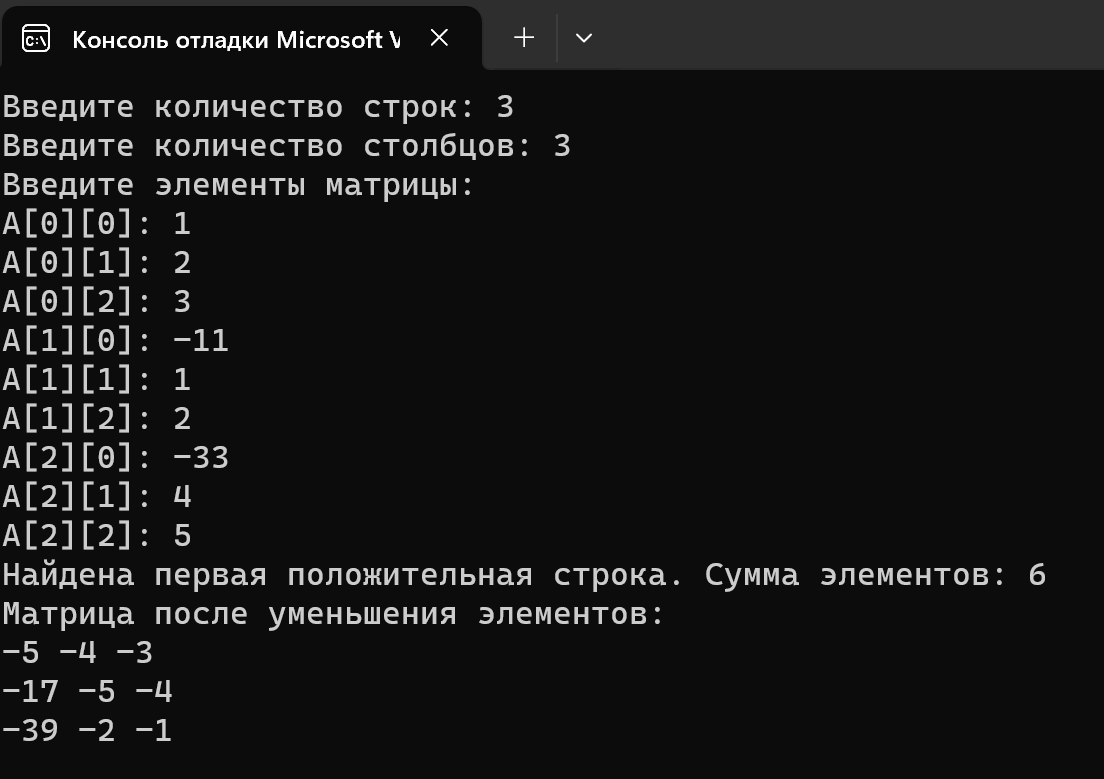
}

// Освобождение выделенной памяти

delete[] matrix;

return 0;

}



Дополнительные задания

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>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "RUS");

const int n = 2; // задаем размер матрицы

int A[n][n];

int B[n][n];

srand(time(NULL));

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

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

A[i][j] = rand() % 21 - 10; // указываем диапазон от -10 до 10

}

}

cout << "Начальная матрица: " << endl;//вывод первоначальной матрицы

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

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

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

}

cout << endl;

}

for (int i = 0; i < n; i += 2) { // переставляем согласно схеме из условия

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

B[i][j] = A[i + 1][j + 1];

B[i + 1][j] = A[i][j + 1];

B[i + 1][j + 1] = A[i][j];

B[i][j + 1] = A[i + 1][j];

}

}

cout << "Новая матрица: " << endl;//вывод новой матрицы

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

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

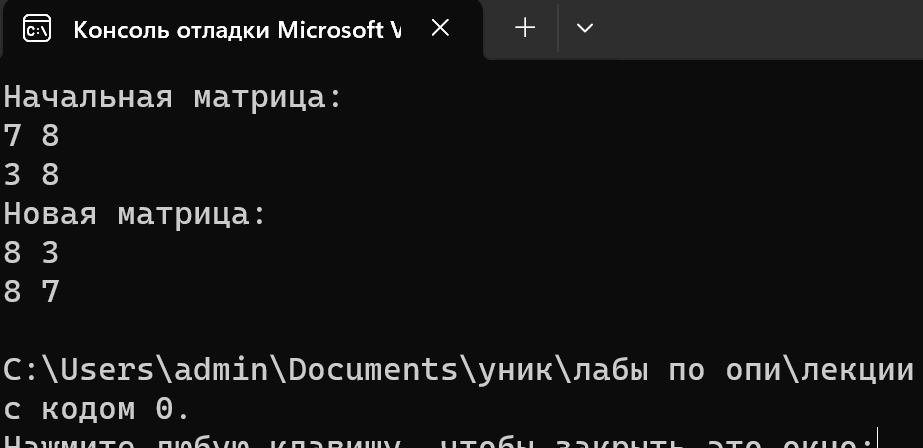
cout << B[i][j] << " ";

}

cout << endl;

}

}



1. 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>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "RUS");

int n, m;

int A[500][500]; //выделяем память под массив

cout << "Введите размер массива: ";

cin >> n;

for (int i = 0; i < n; i++) {// формируем латинский квадрат

m = i + 1;

for (int j = 0; j < n; j++) {//заполняем строку элементами

A[i][j] = m;

m += 1;

if (m > n) {

m = 1;

}

}

}

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

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

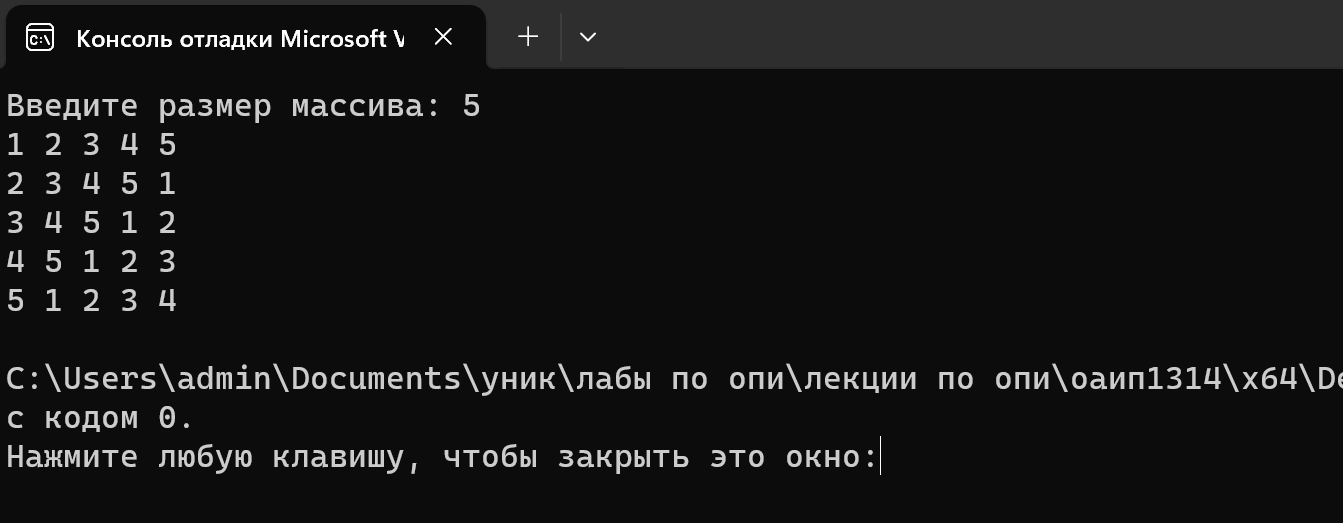
cout << A[i][j] << " ";

}

cout << endl;//выводим в консоль

}

}



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>

#include <algorithm>

using namespace std;

const int N = 3; //указываем размер матрицы

int main() {

setlocale(LC\_CTYPE, "Rus");

double matrix[N][N];

// Заполняем матрицу значениями

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

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

cout << "Введите элемент [" << i << "][" << j << "]: ";

cin >> matrix[i][j];

}

}

// Перестановка элементов по диагонали

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

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

if (i != j) {

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

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

}

}

}

// Вывод преобразованной матрицы

cout << "Переставленная матрица:" << endl;

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

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

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

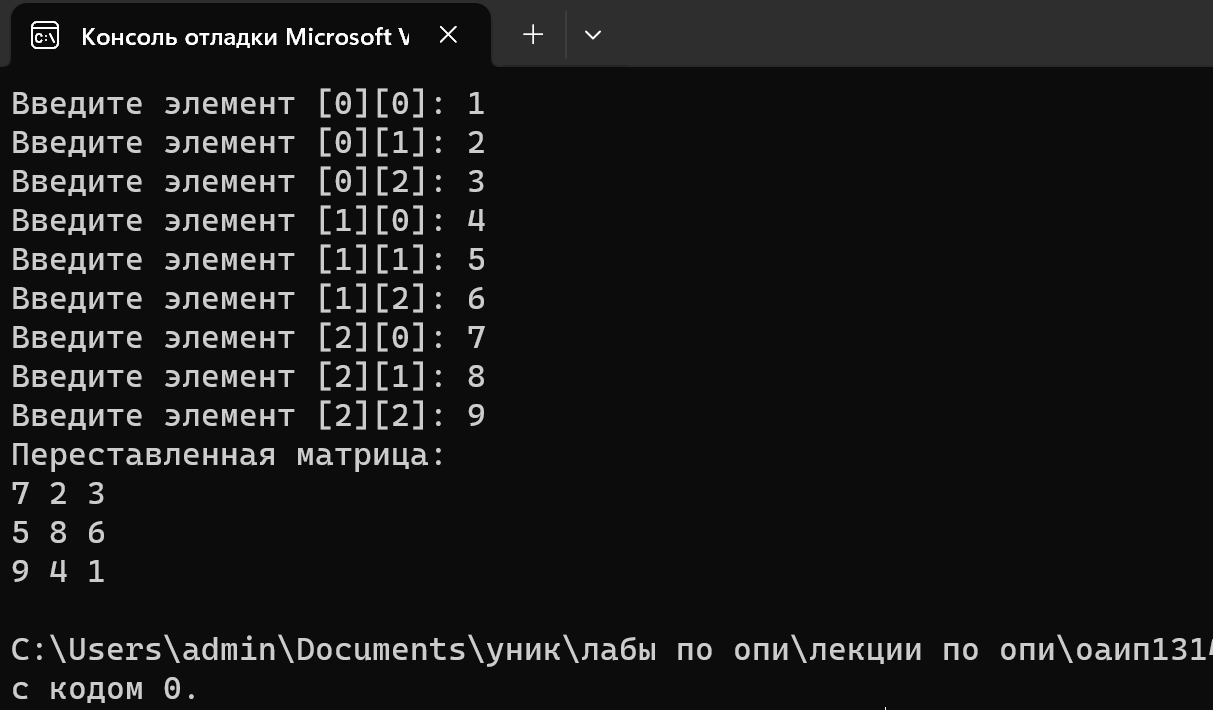
}

cout << endl;

}

return 0;

}

****