**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 14

По дисциплине «Основы алгоритмизации и программирования»

На тему «Многомерные массивы»

Выполнил:  
Студент 1 курса, 10 группа  
Короткевич Артём Сергеевич  
Преподаватель: асс. Харланович А.В

2024, Минск  
  
  
  
  
  
  
**Задание 1**  
  
Выполнить программу, записанную в правой части. Внести изменения в программу с тем, чтобы инициализировался другой массив. Осуществить вывод этого массива в виде матрицы.  
  
**Код**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

const int ci = 3, cj = 2;

int a[ci][cj] = { {1, 2}, {3, 4}, {5, 6} };

cout << "Первый массив (с индексами): " << endl;

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

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

cout << "a[" << i << "," << j << "] = " << a[i][j] << endl;

}

}

int b[ci][cj] = { {7, 8}, {9, 10}, {11, 12} };

cout << "\nВторой массив (в виде матрицы): " << endl;

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

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

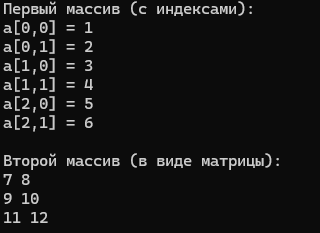
cout << b[i][j] << " ";

}

cout << endl;

}

return 0;   
}

**Результат**  
  
  
 **Задание 2**  
  
Выполнить программы, записанные в правой части для одной и той же задачи, условие которой надо определить. Одна из программ использует указатели.  
  
**Условие**  
Написать два алгоритма, которые будут выводить сумму элементов массива с использованием индекса и указателей.  
  
**Код(1)**

#include <iostream>  
using namespace std;

void main()

{

const int n = 3, m = 2;

int h = 0;

int A[n][m] = { 0, 1, 2, 3, 4, 5 };

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

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

h += A[i][j];

cout << h;   
}

**Результат(1)**  
  
  
  
**Код(2)**

#include <iostream>  
using namespace std;

void main()

{

const int n = 3, m = 2;

int h = 0;

int C[n][m] = { 0, 1, 2, 3, 4, 5 };

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

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

h += \*(\*(C + i) + j);

cout << h;   
}

**Результат(2)**  
  
  
  
**Задание 3**

Изучить способы работы с двумерными массивами, выполнив программу в правой части и записав ее условие.  
Внести изменения в программу с тем, чтобы определялись минимальные элементы каждой строки. Для доступа к элементам матрицы использовать указатели.  
  
**Код**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 2, m = 4; // размер двумерного массива

// объявление двумерного массива

int B[n][m];

int i, j;

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

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

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

cin >> \*(\*(B + i) + j);

}

}

cout << "\nИсходный массив:" << endl;

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

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

cout << "B[" << i << "," << j << "] = " << \*(\*(B + i) + j) << "\t";

}

cout << endl;

}

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

int\* rowStart = \*(B + i); // указатель на начало строки

int min = \*rowStart; // предполагаем, что первый элемент строки будет минимальным

for (j = 1; j < m; j++) { // проходим по строке, начиная со второго элемента

if (\*(rowStart + j) < min) {

min = \*(rowStart + j); // обновление минимального элемента

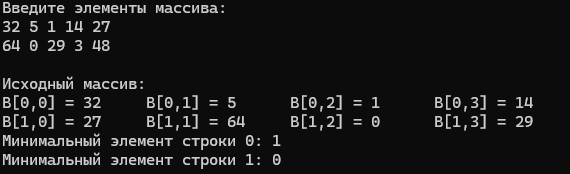
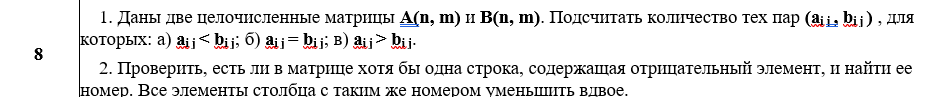
}

}

cout << "Минимальный элемент строки " << i << ": " << min << endl;

}

return 0;   
}

**Результат  
  
**  
**Задание 4**  
  
**Вариант 8  
  
  
  
Код(1)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 3, m = 3; // размеры матриц

int A[n][m] = { {4, 5, 6}, {1, 2, 3}, {7, 8, 9} };

int B[n][m] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

int lessCount = 0, equalCount = 0, greaterCount = 0;

for (int i = 0; i < n; ++i) { // подсчёт количества пар

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

if (A[i][j] < B[i][j]) {

++lessCount;

}

else if (A[i][j] == B[i][j]) {

++equalCount;

}

else {

++greaterCount;

}

}

}

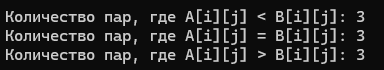
cout << "Количество пар, где A[i][j] < B[i][j]: " << lessCount << endl;

cout << "Количество пар, где A[i][j] = B[i][j]: " << equalCount << endl;

cout << "Количество пар, где A[i][j] > B[i][j]: " << greaterCount << endl;

return 0;

}

**Результат(1)  
  
  
  
Код(2)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 3, m = 3; // размер матрицы

int matrix[n][m] = { {1, -2, 3}, {4, 5, 6}, {7, 8, 9} };

int negativeRow = -1; // номер строки с отрицательным элементом

for (int i = 0; i < n; ++i) { // поиск строки с отрицательным элементом

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

if (matrix[i][j] < 0) {

negativeRow = i;

break;

}

}

if (negativeRow != -1) break; // прекращаем поиск, если строка найдена

}

if (negativeRow != -1) {

cout << "Строка с отрицательным элементом найдена: " << negativeRow + 1 << endl;

for (int i = 0; i < n; ++i) { // уменьшение всех элементов столбца с таким же номером вдвое

matrix[i][negativeRow] /= 2;

}

}

else {

cout << "В матрице нет строк с отрицательными элементами." << endl;

}

cout << "Изменённая матрица:" << endl;

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

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

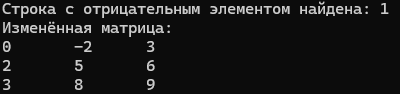
cout << matrix[i][j] << "\t";

}

cout << endl;

}

return 0;  
}

**Результат(2)  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Дополнительные задания  
  
Задание 1  
  
  
  
Код**

#include <iostream>  
using namespace std;

int main()

{

srand(time(NULL)); // инициализация генератора случайных чисел

setlocale(LC\_ALL, "Russian");

const int n = 2; // размер четверти матрицы

const int mas = 2 \* n; // размер длины и ширины матрицы

int A[mas][mas]; // исходная матрица размером masx x mas

cout << "Первоначальная расстановка:" << endl; // вывод первоначальной расстановки

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

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

A[i][j] = rand() % 21 - 10; // заполняем матрицу случайными числами от -10 до 10

cout << A[i][j] << " "; // выводим значение элемента матрицы

}

cout << endl; // переход на новую строку для следующей строки матрицы

}

for (int i = 0; i < n; ++i) { // перестановка блоков в матрице

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

int temp = A[i][j]; // меняем местами блоки в противоположных квадрантах

A[i][j] = A[i + n][j + n];

A[i + n][j + n] = temp;

temp = A[i + n][j]; // меняем местами соседние блоки

A[i + n][j] = A[i][j + n];

A[i][j + n] = temp;

}

}

cout << "После перестановки блоков:" << endl; // вывод матрицы после перестановки блоков

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

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

cout << A[i][j] << " "; // выводим значение элемента матрицы

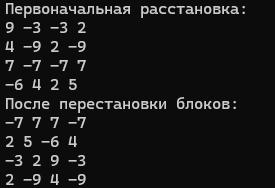
}

cout << endl; // переход на новую строку для следующей строки матрицы

}

return 0;  
}

**Результат**

  
  
**Задание 2  
  
  
  
Код**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL)); // подключаем генератор случайных чисел

//вводим размер матрицы

int n;

cout << "Введите размер матрицы:\n";

cin >> n;

int\*\* A = new int\* [n];// с помощью указателей инициализируем матрицу А

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

A[i] = new int[n]; // выделяем память для каждой строки

}

// выводим получившуюся матрицу, используя генератор случайных чисел

cout << "Матрица: " << endl;

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

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

A[i][j] = rand() % n + 1;

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

}

cout << '\n';

}

// удаляем массив из памяти

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

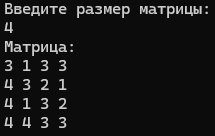
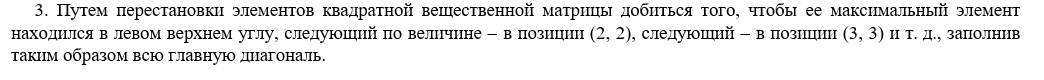
delete[] A[i];

}

delete[] A;

return 0;

}

**Результат  
  
  
  
  
Задание 3  
  
  
  
Код**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));//подключаем генератор случайных чисел

//инициализируем переменные для размера массива, матрицы, максимальных значений и мест расположения в матрице

const int n = 3, m = 3;

int A[n][m];

int max1 = 0, max2 = 0, max3 = 0, row1 = 0, row2 = 0, row3 = 0, column1 = 0, column2 = 0, column3 = 0;

//используем цикл для генерирования элементов и поиска среди них 3 максимальных

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

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

A[i][j] = rand() % 100;

if (A[i][j] > max1) {

max3 = max2; row3 = row2; column3 = column2;

max2 = max1; row2 = row1; column2 = column1;

max1 = A[i][j]; row1 = i; column1 = j;

}

else if (A[i][j] > max2) {

max3 = max2; row3 = row2; column3 = column2;

max2 = A[i][j]; row2 = i; column2 = j;

}

else if (A[i][j] > max3) {

max3 = A[i][j]; row3 = i; column3 = j;

}

}

}

//перестановка элементов с помощью временной переменной

int temp = A[0][0];

A[0][0] = max1;

A[row1][column1] = temp;

temp = A[n / 2][m / 2];

A[n / 2][m / 2] = max2;

A[row2][column2] = temp;

temp = A[n - 1][m - 1];

A[n - 1][m - 1] = max3;

A[row3][column3] = temp;

//выводим новую матрицу

cout << "Получившаяся матрица:" << endl;

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

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

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

}

cout << endl;

}

return 0;

}

**Результат**  
