МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

Программирование

Лабораторная работа № 5 (часть 2)

Тема

Изучение приемов работы с многомерными массивами в языке C++.

Выполнил

Студент гр. Б260221

Воробьёв А.А.

Проверила

Сафронова М.А.

Тула 2023 г.

**Цель работы:**

Задача лабораторной работы по C++ на тему "Изучение приемов работы с многомерными массивами в языке C++" заключается в том, чтобы овладеть навыками работы с многомерными массивами в языке C++. Многомерные массивы представляют собой структуру данных, которая позволяет хранить элементы в виде таблицы или матрицы, имеющей две или более размерности.

В рамках данной лабораторной работы целью является достижение следующих результатов:

1. Понимание синтаксиса объявления и инициализации многомерных массивов в C++. Изучение способов создания двумерных, трехмерных и многомерных массивов.
2. Овладение навыками доступа к элементам многомерных массивов. Изучение индексации и использование циклов для обхода элементов многомерных массивов.
3. Изучение приемов работы с многомерными массивами, включая манипуляции с элементами, изменение значений, поиск минимальных и максимальных значений, сортировку и другие операции.
4. Овладение навыками работы с многомерными массивами в функциях. Изучение передачи многомерных массивов в функции, возврата массивов из функций и обработки многомерных массивов внутри функций.
5. Изучение особенностей работы с динамическими многомерными массивами. Овладение навыками выделения и освобождения памяти для многомерных массивов.
6. Применение изученных приемов работы с многомерными массивами для решения различных задач, таких как поиск суммы элементов, поиск определенного значения, транспонирование матрицы и других манипуляций с данными.

**Задание 1. Основные операции при работе с массивами**

Разработать программу на языке C++ для решения следующей задачи.

В двумерном массиве B, состоящем из n × m вещественных чисел, вычислить:

• произведение элементов;

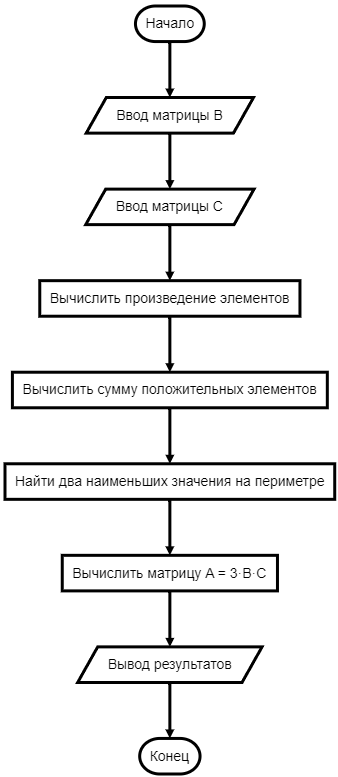
• сумму положительных элементов;

• два наименьших значения среди элементов, расположенных по периметру матрицы.

Для заданной матрицы B(n×m) и матрицы того же типа, но другой размерности

C(m×n) найти значение выражения A=3·B·C

**Блок-схема:**



**Код программы:**

void Task1()

{

int n, m;

std::cout << "Введите количество строк и столбцов матрицы B: ";

std::cin >> n >> m;

std::vector<std::vector<double>> B = InputMatrix(n, m);

std::cout << "Введите количество строк и столбцов матрицы C: ";

std::cin >> n >> m;

std::vector<std::vector<double>> C = InputMatrix(n, m);

// your logic for Task1

double product = 1;

double sumPositive = 0;

double min1 = B[0][0], min2 = B[0][0];

for (int i = 0; i < B.size(); i++) {

for (int j = 0; j < B[i].size(); j++) {

product \*= B[i][j];

if (B[i][j] > 0) sumPositive += B[i][j];

if (i == 0 || i == B.size() - 1 || j == 0 || j == B[i].size() - 1) {

if (B[i][j] < min1) {

min2 = min1;

min1 = B[i][j];

} else if (B[i][j] < min2) {

min2 = B[i][j];

}

}

}

}

std::vector<std::vector<double>> A = matrixMult(B, C);

for (auto &row : A) {

for (auto &el : row) {

el \*= 3;

}

}

std::cout << "Произведение элементов: " << product << std::endl;

std::cout << "Сумма положительных элементов: " << sumPositive << std::endl;

std::cout << "Два наименьших значения на периметре: " << min1 << ", " << min2 << std::endl;

std::cout << "Матрица A = 3·B·C: " << std::endl;

for (const auto &row : A) {

for (const auto &el : row) {

std::cout << el << ' ';

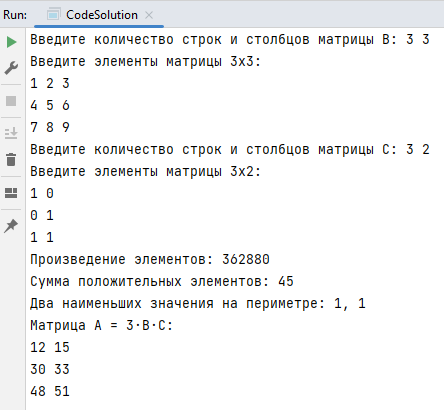
}

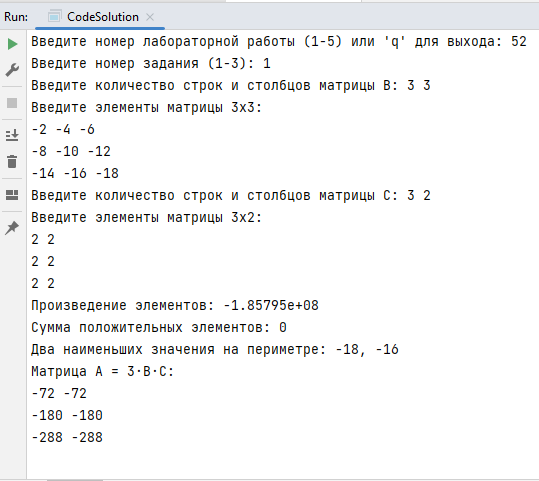
std::cout << '\n';

}

}

**Вывод программы:**



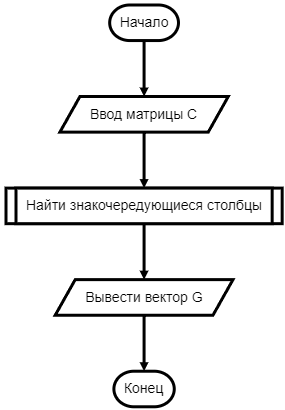


**Задание 2. Работа со строками и столбцами матрицы**

Разработать программу на языке C++ для решения следующей задачи.

Задана матрица вещественных чисел C(k × m). Сформировать вектор D(k) из средних арифметических положительных значений строк матрицы, и вектор G(n) из номеров столбцов, которые представляют собой знакочередующийся ряд.

**Блок-схема:**

****

**Код программы:**

void Task2()

{

int n, m;

std::cout << "Введите количество строк и столбцов матрицы C: ";

std::cin >> n >> m;

std::vector<std::vector<double>> C = InputMatrix(n, m);

// your logic for Task2

std::vector<double> D(C.size());

std::vector<int> G;

for (int i = 0; i < C.size(); i++) {

double sumPositive = 0;

int countPositive = 0;

for (int j = 0; j < C[i].size(); j++) {

if (C[i][j] > 0) {

sumPositive += C[i][j];

countPositive++;

}

}

D[i] = countPositive > 0 ? sumPositive / countPositive : 0;

}

for (int j = 0; j < C[0].size(); j++) {

bool alternatingSigns = true;

for (int i = 1; i < C.size() && alternatingSigns; i++) {

if (std::signbit(C[i][j]) == std::signbit(C[i - 1][j])) {

alternatingSigns = false;

}

}

if (alternatingSigns) {

G.push\_back(j);

}

}

std::cout << "Вектор D: ";

for (double el : D) {

std::cout << el << " ";

}

std::cout << std::endl;

std::cout << "Вектор G: ";

for (int col : G) {

std::cout << col << " ";

}

std::cout << std::endl;

}

std::vector<std::vector<double>> InputMatrix(int n, int m) {

std::vector<std::vector<double>> matrix(n, std::vector<double>(m));

std::cout << "Введите элементы матрицы " << n << "x" << m << ":\n";

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

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

std::cin >> matrix[i][j];

}

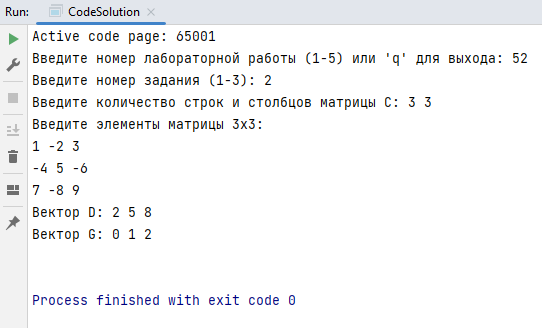
}

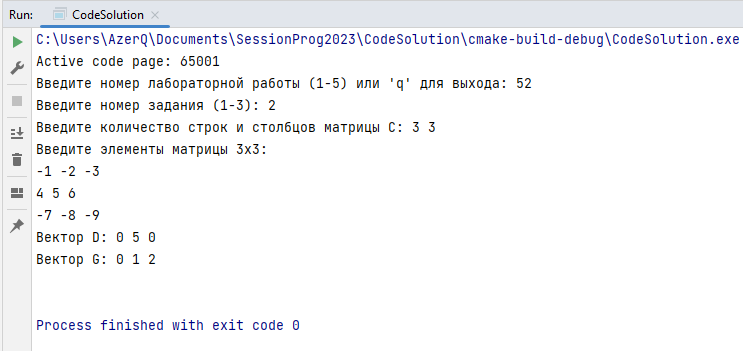
return matrix;

}

}

**Вывод программы:**





**Вывод**

Лабораторная работа "Изучение приемов работы с многомерными массивами в языке C++" позволяет овладеть навыками работы с многомерными массивами в языке C++ и применять их для эффективного решения задач, связанных с обработкой данных в виде матриц и таблиц.

В ходе выполнения данной лабораторной работы были достигнуты следующие результаты:

1. Изучены основные принципы объявления и инициализации многомерных массивов в языке C++. Были рассмотрены способы создания двумерных, трехмерных и многомерных массивов.
2. Освоены навыки доступа к элементам многомерных массивов, включая индексацию и использование циклов для обхода элементов.
3. Изучены приемы работы с многомерными массивами, такие как манипуляции с элементами, изменение значений, поиск минимальных и максимальных значений, сортировка и другие операции.
4. Освоены навыки работы с многомерными массивами в функциях, включая передачу массивов в функции, возврат массивов из функций и обработку многомерных массивов внутри функций.
5. Изучены особенности работы с динамическими многомерными массивами, включая выделение и освобождение памяти.
6. Применены изученные приемы работы с многомерными массивами для решения различных задач, включая поиск суммы элементов, поиск определенного значения, транспонирование матрицы и другие манипуляции с данными.

Лабораторная работа помогла углубить понимание работы с многомерными массивами в языке C++ и развить навыки эффективной обработки данных в виде матриц и таблиц. Полученные знания и навыки могут быть применены в различных областях программирования, где требуется работа с многомерными данными.