Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«**Информационные технологии и автоматизированные системы**»**

направление подготовки: 09.03.04 – **«**Программная инженерия»

**Лабораторная работа №5.**

**“Функции и массивы”  
Вариант 10**

Выполнил студент гр. РИС-24-2б

Бартов Игорь Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Работа с одномерными массивами**

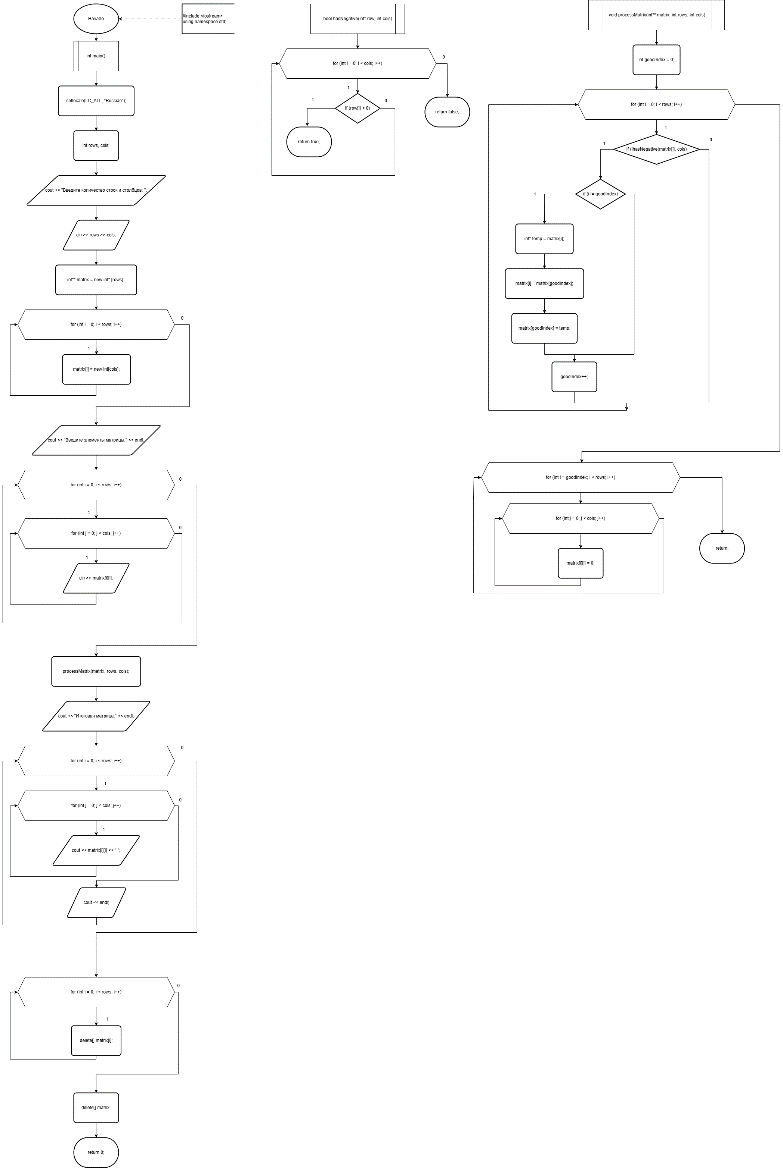
1. **Постановка задачи:**

Постановка задачи: написать функцию, проверяющую есть ли отрицательные элементы в указанной строке двумерного массива. Удалить из массива все строки с отрицательными элементами, удаленная строк заполняется 0 и переноситься в конец массива.

Анализ решения:

1. Для каждой строки перебираем все элементы. Если находится хотя бы один отрицательный элемент, строка считается «содержащей отрицательные элементы».
2. Проходим по всем строкам и определяем, какие строки не содержат отрицательных элементов («хорошие» строки).
3. Перемещаем «хорошие» строки в начало массива, сохраняя их относительный порядок, используя индекс для следующей свободной позиции.
4. После этого оставшиеся строки считаются «удалёнными» – их содержимое заменяется нулями, а сами строки остаются в массиве, но переносятся в его конец.
5. Массив, в котором сначала идут строки без отрицательных элементов, а затем – строки с отрицательными элементами, заполненные нулями

**2. Блок-схема:**

****

1. **Код:**

#include <iostream>

using namespace std;

bool hasNegative(int\* row, int cols) {

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

if (row[i] < 0)

return true;

return false;

}

void processMatrix(int\*\* matrix, int rows, int cols) {

int goodIndex = 0;

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

if (!hasNegative(matrix[i], cols)) {

if (i != goodIndex) {

int\* temp = matrix[i];

matrix[i] = matrix[goodIndex];

matrix[goodIndex] = temp;

}

goodIndex++;

}

}

for (int i = goodIndex; i < rows; i++)

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

matrix[i][j] = 0;

}

int main() {

setlocale(LC\_ALL, "Russian");

int rows, cols;

cout << "Введите количество строк и столбцов: ";

cin >> rows >> cols;

int\*\* matrix = new int\* [rows];

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

matrix[i] = new int[cols];

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

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

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

cin >> matrix[i][j];

processMatrix(matrix, rows, cols);

cout << "Итоговая матрица:" << endl;

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

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

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

cout << endl;

}

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

delete[] matrix[i];

delete[] matrix;

return 0;

}