Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №5**

Дисциплина: основы алгоритмизации и программирования

Тема: функции и массивы

Вариант 23

Выполнил:

Студент группы ИВТ-20-2б

Брейкин Алексей Дмитриевич

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2020**

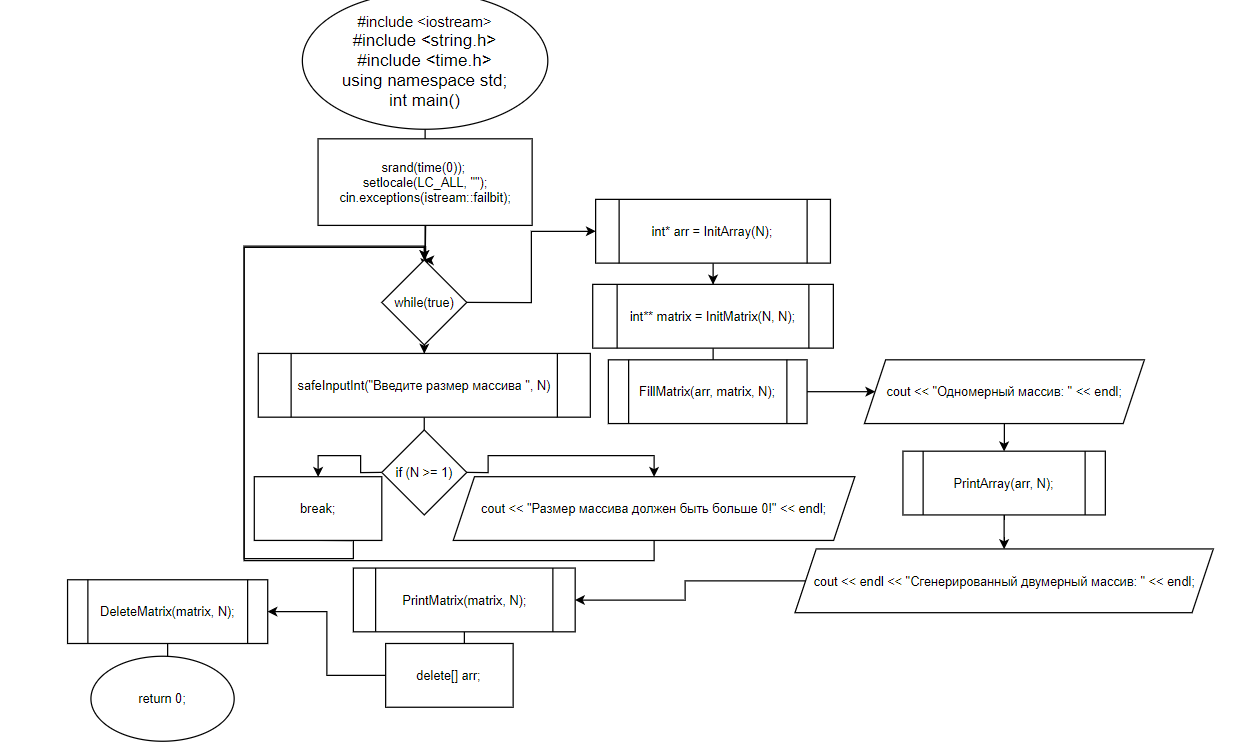
**Постановка задачи**

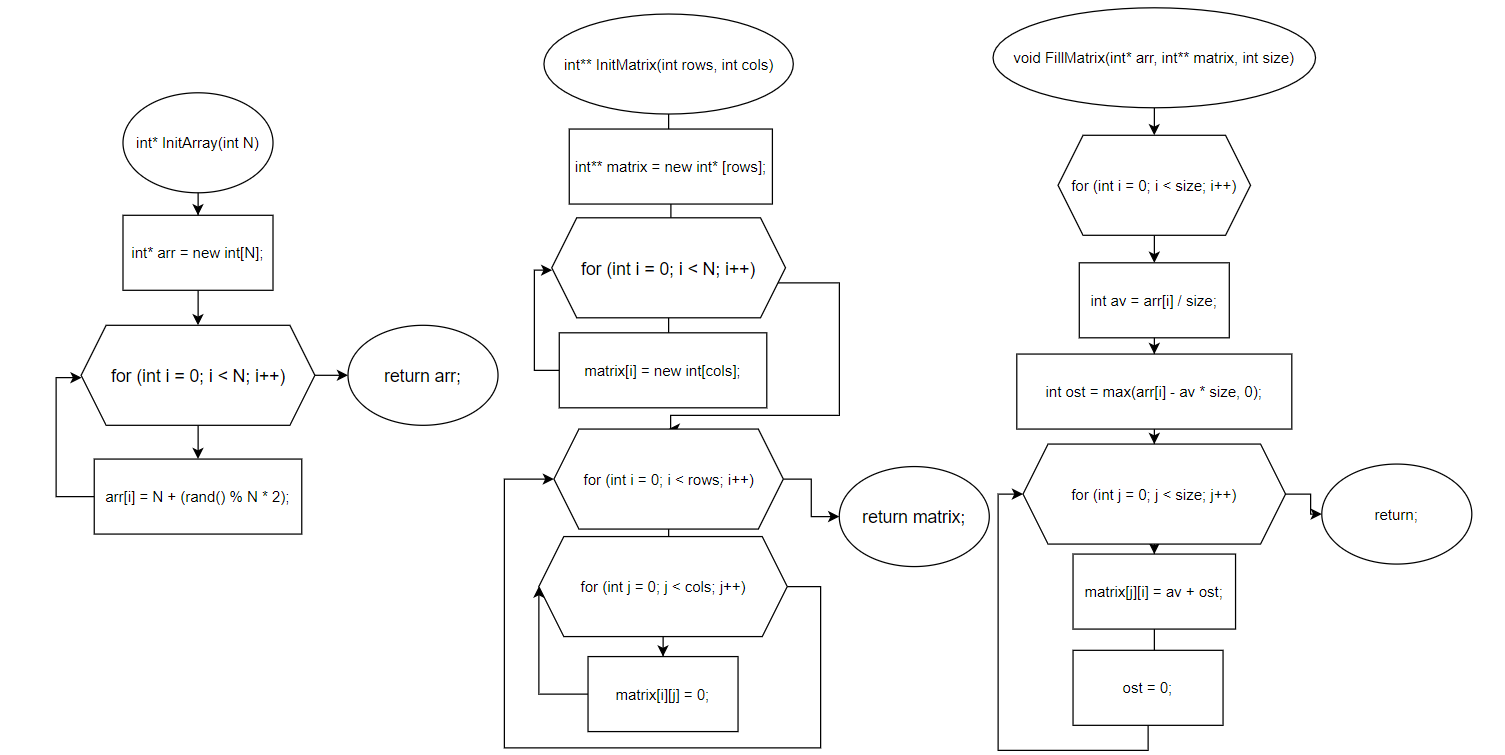
Задан одномерный массив, состоящий из N целых чисел. Сформировать на его основе двумерный массив N x N так, чтобы сумма элементов в первом столбце была равна первому элементу одномерного массива, сумма элементов во втором столбце была равна второму элементу одномерного массива и т. д. Нули не использовать.

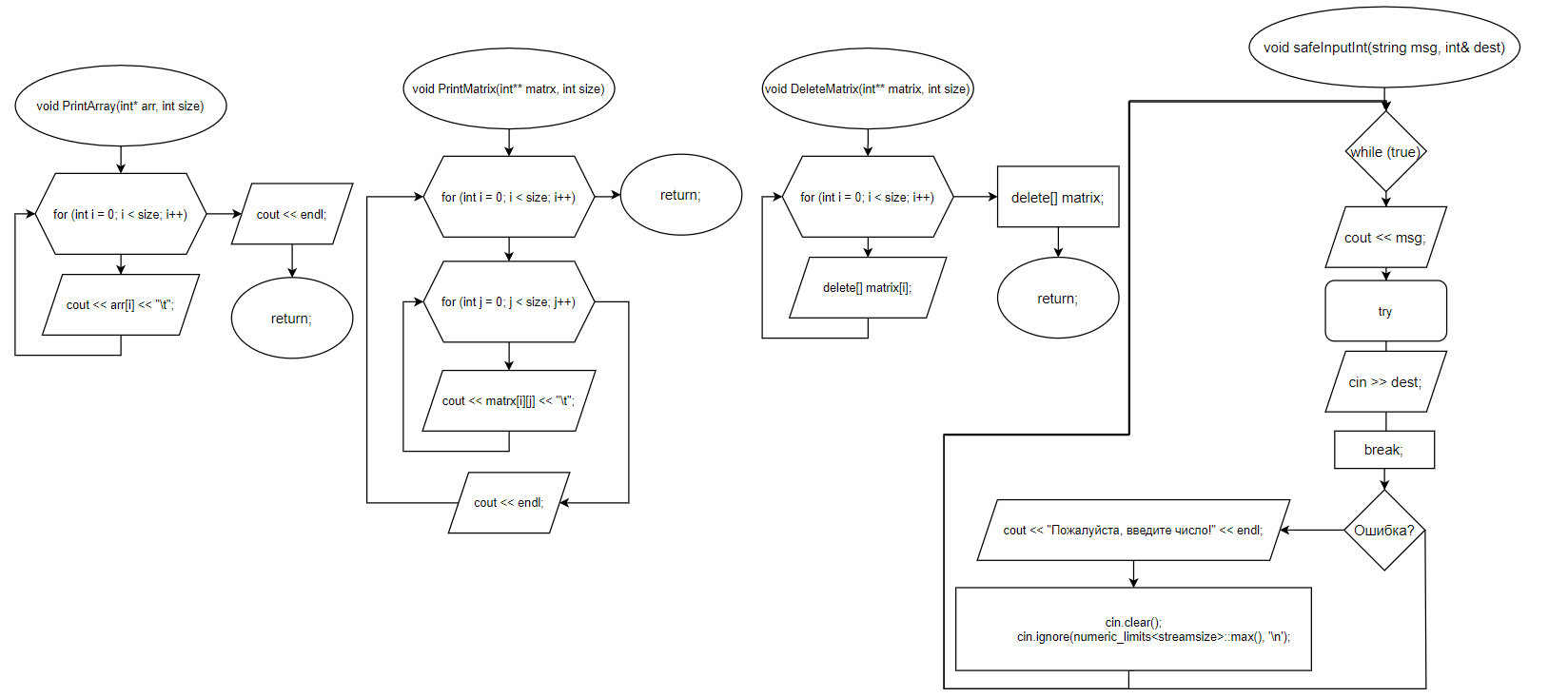
**Анализ задачи**

1. Запрашиваем пользователя число N и создаём одномерный динамический массив на N целых чисел и с помощью датчика случайных чисел заполняем его.
2. Создаём двумерный массив размером N x N, используя массив указателей на указатели.
3. Заполняем матрицу с помощью вложенного цикла. Для этого мы берём значение из одномерного массива, индекс которого равен столбцу матрицы, и делим его на количество столбцов. Сохраняем остаток и добавляем его к первому элементу первой строки.
4. Выводим на экран полученный одномерный массив и матрицу с помощью циклов.
5. Очищаем память, что мы выделили под одномерный и двумерный массивы.

**Блок-схема**







**Исходный код**

#include <iostream>

#include <string.h>

#include <time.h>

using namespace std;

int\* InitArray(int N) {

int\* arr = new int[N];

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

arr[i] = N + (rand() % N \* 2);

}

return arr;

}

int\*\* InitMatrix(int rows, int cols) {

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

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

matrix[i] = new int[cols];

}

// init the matrix with zero

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

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

matrix[i][j] = 0;

}

}

return matrix;

}

void FillMatrix(int\* arr, int\*\* matrix, int size) {

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

int av = arr[i] / size;

int ost = max(arr[i] - av \* size, 0);

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

matrix[j][i] = av + ost;

ost = 0;

}

}

}

void PrintArray(int\* arr, int size) {

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

cout << arr[i] << "\t";

}

cout << endl;

}

void PrintMatrix(int\*\* matrx, int size) {

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

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

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

}

cout << endl;

}

}

void DeleteMatrix(int\*\* matrix, int size) {

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

delete[] matrix[i];

}

delete[] matrix;

}

void safeInputInt(string msg, int& dest) {

for (;;) {

cout << msg;

try {

cin >> dest;

break;

}

catch (std::ios\_base::failure const& ex) {

cout << "Пожалуйста, введите число!" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

int main() {

srand(time(0));

setlocale(LC\_ALL, "");

cin.exceptions(istream::failbit);

int N;

for (;;) {

safeInputInt("Введите размер массива: ", N);

if (N >= 1) break;

cout << "Размер массива должен быть больше 0!" << endl;

}

int\* arr = InitArray(N);

int\*\* matrix = InitMatrix(N, N); // matrix NxN -- square matrix

FillMatrix(arr, matrix, N);

cout << "Одномерный массив: " << endl;

PrintArray(arr, N);

cout << endl << "Сгенерированный двумерный массив: " << endl;

PrintMatrix(matrix, N);

delete[] arr;

DeleteMatrix(matrix, N);

return 0;

}

**Скриншоты работы программы**

