Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа

**“** **Перегрузка функций в C++”**

Выполнил:

студент группы РИС-23-1б

Жуланов Никита Андреевич

Проверила:

доцент кафедры ИТАС

О. А. Полякова

2024 г.

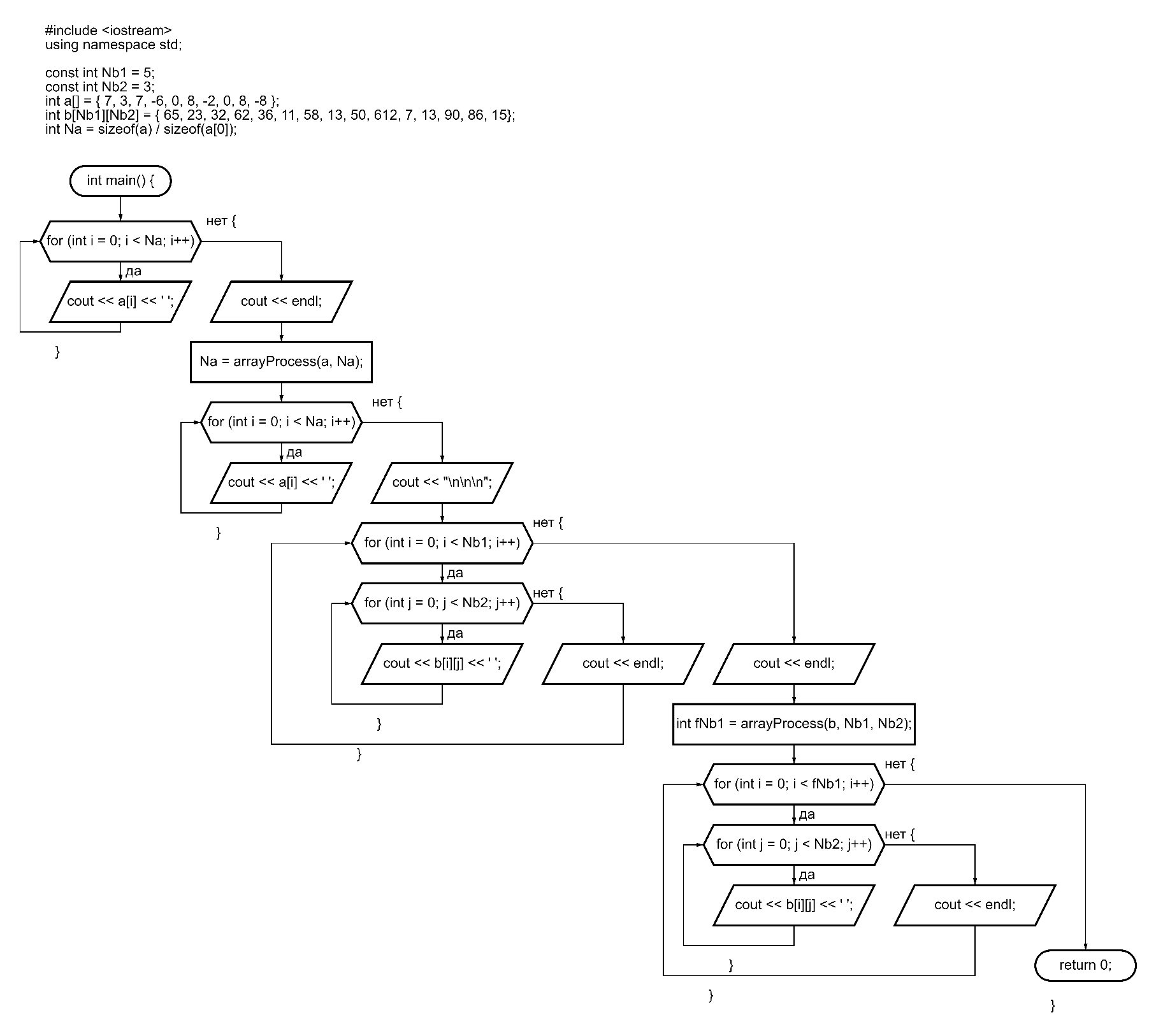
**Разработка алгоритма** **с перегруженной функцией**

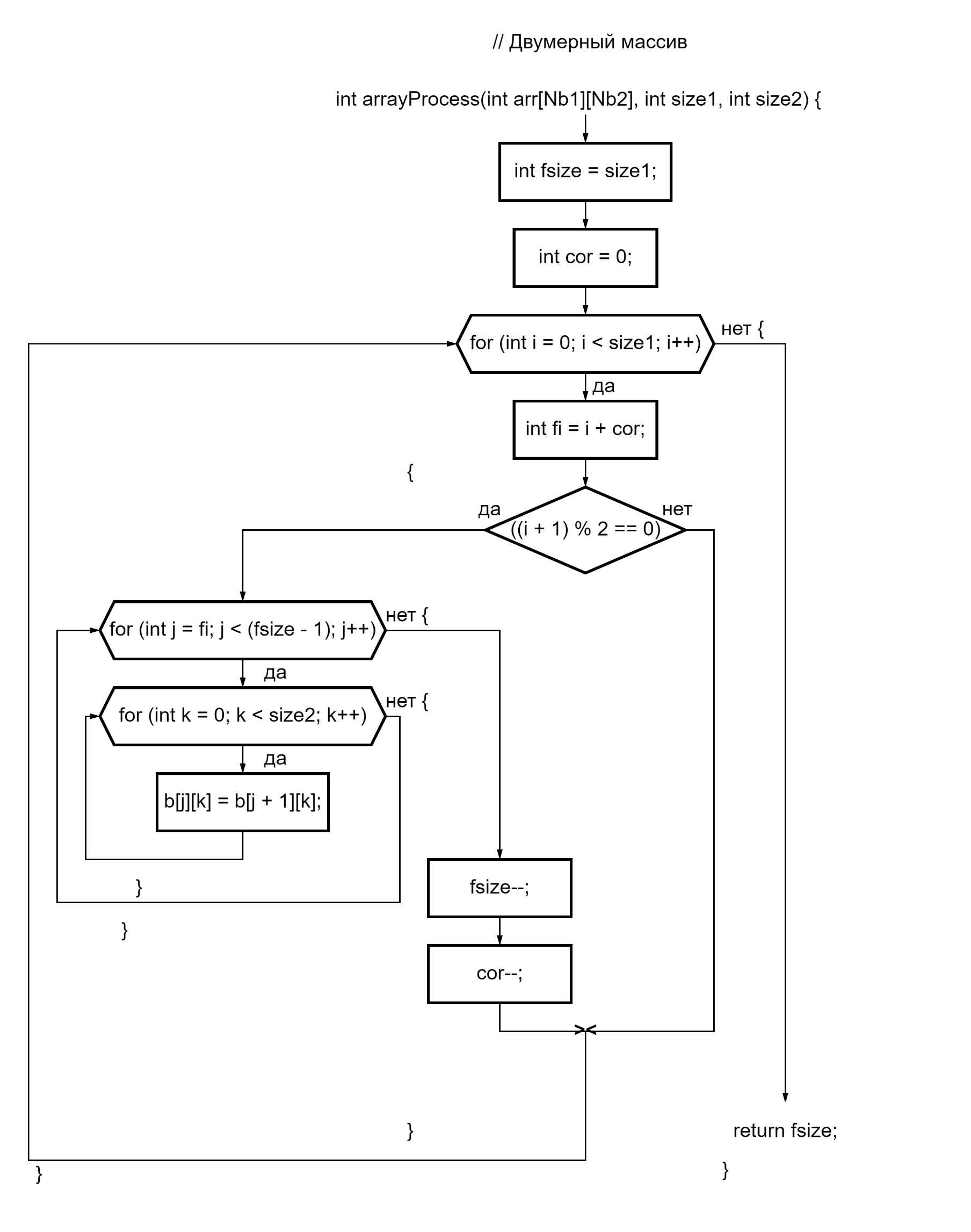
**Постановка задачи:** написать перегруженные функции и основную программу, которая их вызывает.

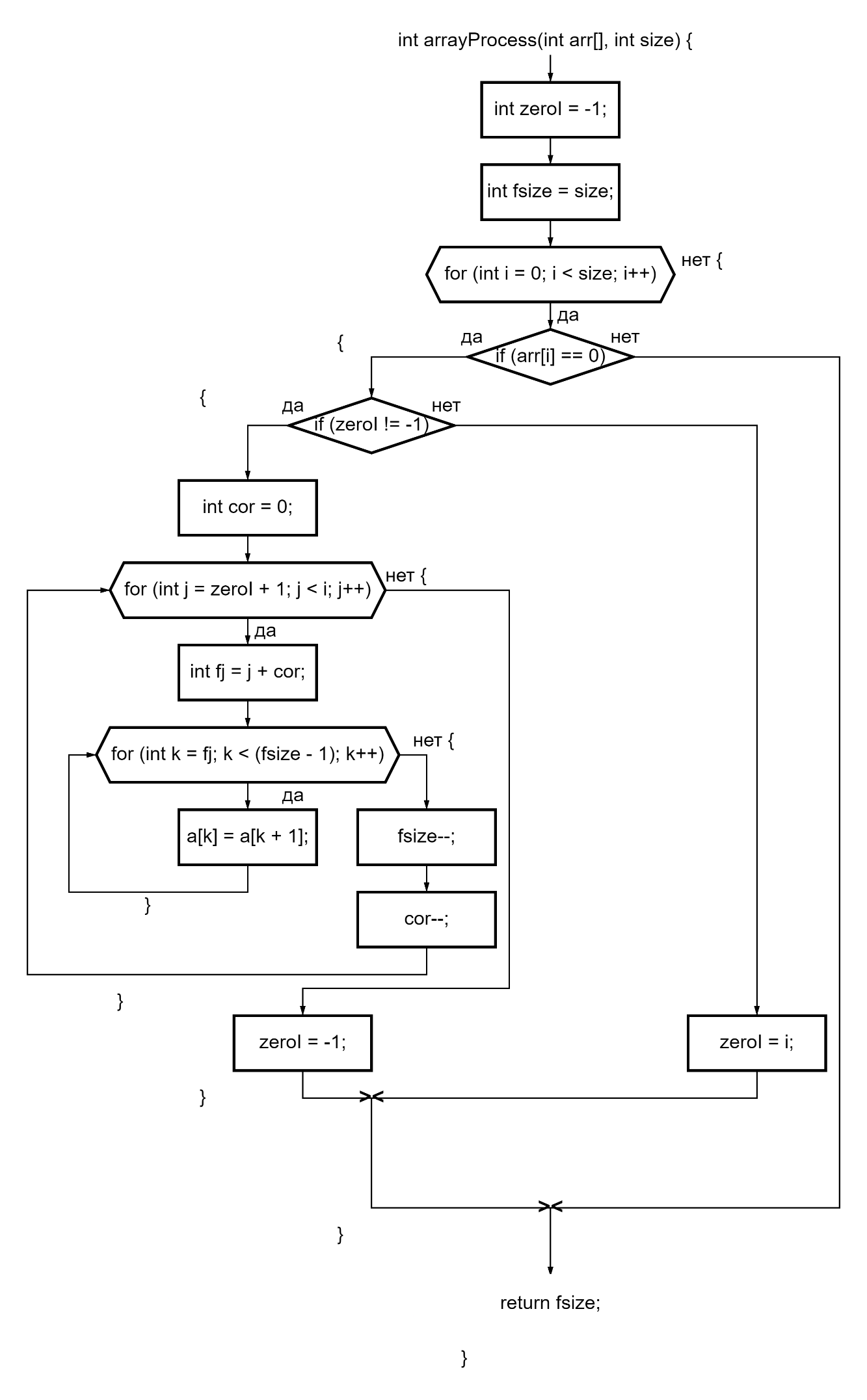
**Вариант задания - 25:**

а) для двумерного массива удаляет все четные строки;

б) для одномерного массива удаляет все элементы, заключённые между двумя нулевыми элементами.

**Блок-схемы:**

**Алгоритм для двумерного массива:**

**Алгоритм для одномерного массива:**

**Код на языке C++:**

#include <iostream>

using namespace std;

const int Nb1 = 5;

const int Nb2 = 3;

int a[] = { 7, 3, 7, -6, 0, 8, -2, 0, 8, -8 };

int b[Nb1][Nb2] = { 65, 23, 32, 62, 36, 11, 58, 13, 50, 612, 7, 13, 90, 86, 15};

int Na = sizeof(a) / sizeof(a[0]);

int arrayProcess(int arr[Nb1][Nb2], int size1, int size2) {     // Двумерный массив

    int fsize = size1;

    int cor = 0;

    for (int i = 0; i < size1; i++) {

        int fi = i + cor;

        if ((i + 1) % 2 == 0) {

            for (int j = fi; j < (fsize - 1); j++) {

                for (int k = 0; k < size2; k++) {

                    b[j][k] = b[j + 1][k];

                }

            }

            fsize--;

            cor--;

        }

    }

    return fsize;

}

int arrayProcess(int arr[], int size) {                         // Одномерный массив

    int zeroI = -1;

    int fsize = size;

    for (int i = 0; i < size; i++) {

        if (arr[i] == 0) {

            if (zeroI != -1) {

                int cor = 0;

                for (int j = zeroI + 1; j < i; j++) {

                    int fj = j + cor;

                    for (int k = fj; k < (fsize - 1); k++) {

                        a[k] = a[k + 1];

                    }

                    fsize--;

                    cor--;

                }

                zeroI = -1;

            }

            else {

                zeroI = i;

            }

        }

    }

    return fsize;

}

int main() {

    for (int i = 0; i < Na; i++) {

        cout << a[i] << ' ';

    }

    Na = arrayProcess(a, Na);

    cout << endl;

    for (int i = 0; i < Na; i++) {

        cout << a[i] << ' ';

    }

    cout << "\n\n\n";

    for (int i = 0; i < Nb1; i++) {

        for (int j = 0; j < Nb2; j++) {

            cout << b[i][j] << ' ';

        }

        cout << endl;

    }

    cout << endl;

    int fNb1 = arrayProcess(b, Nb1, Nb2);

    for (int i = 0; i < fNb1; i++) {

        for (int j = 0; j < Nb2; j++) {

            cout << b[i][j] << ' ';

        }

        cout << endl;

    }

    return 0;

}

**Результаты:**

|  |
| --- |
|  |

**Вывод:**

Я смог реализовать алгоритм с использованием перегруженных функций.