

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 10

Сортировка массивов

Вариант № 12

Преподаватель

подпись, дата

Пушкарев К. В.

Студент КИ18-096, 031830645

_____ 25.11.2018

подпись

Котов С.А.

Красноярск 2018

1 Дополнительное упражнение № 3

Разработать, отладить, продемонстрировать и защитить преподавателю графическую схему алгоритма и составить программу, решающую следующую задачу: на основе исходных целочисленных массивов $X[n]$ и $Y[m]$ сформировать массив C по правилам, представленным ниже, полученный массив отсортировать по невозрастанию. Провести анализ разработанной программы согласно критериям качества.

Правила формирования массива C задания 2: из чисел, которые входят в массив X , но при этом не входят в массив Y (без повторения).

Дополнительные указания задания 2: оформить как функцию поиск заданного числа в массиве

2 Цель работы

Освоение основных приемов обработки массивов, методов доступа к элементам массивов, их реорганизации и модификации.

3 Графическая схема алгоритма

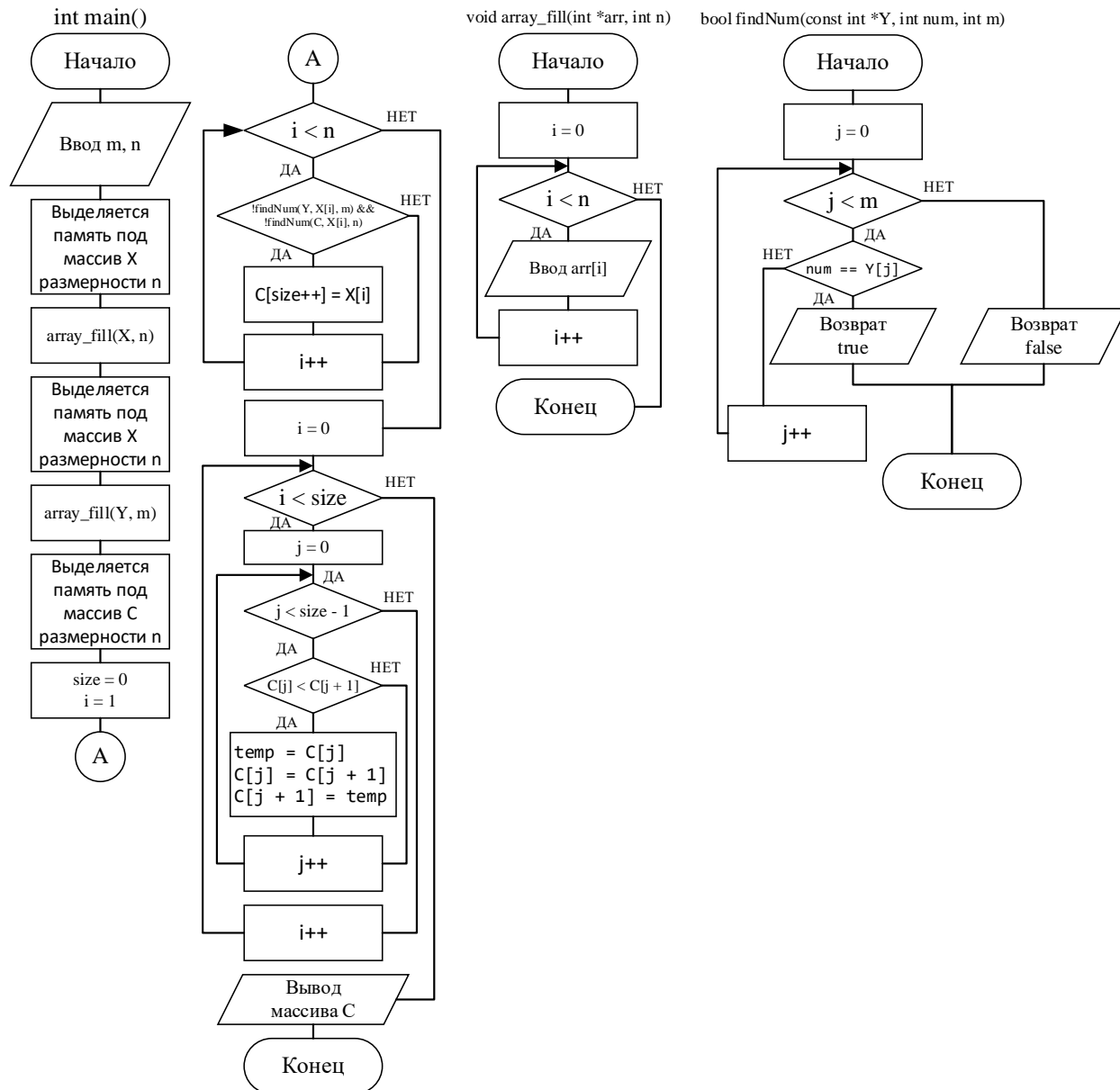


Рисунок 1 – Блок-схема дополнительного упражнения № 3.

4 Код программы

```

1 #include "pch.h"
2 #include <iostream>
3 #include <locale>
4
5 using namespace std;
6
7 // Функция поиска неповторяющихся чисел
8 bool findNum(const int *Y, int num, int m) {
9     for (int j = 0; j < m; j++) {
10         if (num == Y[j]) {
11             return true;
12         }
13     }
14     return false;
  
```

```

15 }
16
17 // Функция заполнения массива
18 void array_fill(int *arr, int n) {
19     for (int i = 0; i < n; i++) {
20         cin >> arr[i];
21     }
22 }
23
24 int main() {
25     setlocale(LC_ALL, "");
26
27     int n, m;
28     cout << "Введите размерность n: ";
29     cin >> n;
30     cout << "Введите размерность m: ";
31     cin >> m;
32
33     int *X = new int[n];
34     cout << "Заполните массив X[n]: ";
35     array_fill(X, n);
36     int *Y = new int[m];
37     cout << "Заполните массив Y[m]: ";
38     array_fill(Y, m);
39
40     // Заполняем массив C
41     int size = 0;
42     int *C = new int[n];
43     for (int i = 1, last = 0; i < n; i++) {
44         if (!findNum(Y, X[i], m) && !findNum(C, X[i], n)) {
45             C[size++] = X[i];
46         }
47     }
48     // Сортируем массив C по убыванию
49     for (int i = 1; i < size; ++i) {
50         for (int j = 0; j < size - i; j++) {
51             if (C[j] < C[j + 1]) {
52                 int temp = C[j];
53                 C[j] = C[j + 1];
54                 C[j + 1] = temp;
55             }
56         }
57     }
58     // Выводим массив C в консоль
59     cout << "Новый массив C: ";
60     for (int i = 0; i < size; i++) {
61         cout << " " << C[i];
62     }
63     cout << endl;
64     return 0;
65 }
66

```

5 Результат выполнения экспериментальной части работы.

Результаты запуска программы с различными входными значениями приведены в таблице 1.

Таблица 1 – Работа программы на тестовых наборах данных

Входные данные	Результат
$n = 5$ $m = 2$ $X = \{2, 3, 3, 2, 4\}$ $Y = \{6, 5\}$	Новый массив C: 4 3 2
$n = 5$ $m = 3$ $X = \{9, 5, 4, 5, 9\}$ $Y = \{4, 8, 2\}$	Новый массив C: 9 5