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 \_\_\_\_\_\_\_\_ Пушкарев К. В.

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_ 25.11.2018 Котов С.А.

подпись

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 Х[n] и Y [m] сформировать массив C по правилам, представленым ниже, полученный массив отсортировать по невозрастанию. Провести анализ разработанной программы согласно критериям качества.  
 Правила формирования массива С задания 2: из чисел, которые входят в массив Х, 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  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | #include "pch.h"  #include <iostream>  #include <clocale>  using namespace std;  // Функция поиска неповторяющихся чисел  bool findNum(const int \*Y, int num, int m) {  for (int j = 0; j < m; j++) {  if (num == Y[j]) {  return true;  }  }  return false;  }  // Функция заполнения массива  void array\_fill(int \*arr, int n) {  for (int i = 0; i < n; i++) {  cin >> arr[i];  }  }  int main() {  setlocale(LC\_ALL, "");  int n, m;  cout << "Введите размерность n: ";  cin >> n;  cout << "Введите размерность m: ";  cin >> m;  int \*X = new int[n];  cout << "Заполните массив X[n]: ";  array\_fill(X, n);  int \*Y = new int[m];  cout << "Заполните массив Y[m]: ";  array\_fill(Y, m);  // Заполняем массив C  int size = 0;  int \*C = new int[n];  for (int i = 1, last = 0; i < n; i++) {  if (!findNum(Y, X[i], m) && !findNum(C, X[i], n)) {  C[size++] = X[i];  }  }  // Сортируем массив C по убыванию  for (int i = 1; i < size; ++i) {  for (int j = 0; j < size - i; j++) {  if (C[j] < C[j + 1]) {  int temp = C[j];  C[j] = C[j + 1];  C[j + 1] = temp;  }  }  }  // Выводим массив С в консоль  cout << "Новый массив C: ";  for (int i = 0; i < size; i++) {  cout << " " << C[i];  }  cout << endl;  return 0;  } |

**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 |