**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 12-13**

**Дисциплина:** Основы алгоритмизации и программирования.

**Тема:** Алгоритм сортировки «выбором»

**Выполнила:**

**студентка группы 201-723**

Круглова А.М.

04.12.20  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил:** преп. Хуснулина Д.Р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2020**

**Оглавление**

[Цель 3](#_Toc58165542)

[Постановка задачи 3](#_Toc58165543)

[Идея алгоритма 3](#_Toc58165544)

[Словесное представление алгоритма 4](#_Toc58165545)

[Блок-схема с использованием элемента модификации 5](#_Toc58165546)

[Листинг программы с использованием параметрического цикла 6](#_Toc58165547)

[Результат работы 1 программы 7](#_Toc58165548)

[Блок-схема без использования элемента модификации 8](#_Toc58165549)

[Листинг программы с использованием цикла с предусловием 9](#_Toc58165550)

[Результат работы 2 программы 10](#_Toc58165551)

# **Цель**

Получить практические навыки разработке алгоритмов и их программной

реализации.

# **Постановка задачи**

Необходимо выполнить и оформить описание следующих пунктов:

1. Сформулировать идею алгоритма
2. Выполнить словесное представление алгоритма
3. Выполнить полнить представление алгоритма с помощью блок схем с использованием элемента модификации и без него.
4. Выполнить программную реализацию алгоритмов на языке С с использованием параметрического цикла и цикла с предусловием.

# **Идея алгоритма**

Сортировка выбором (Selection sort) — алгоритм сортировки. Может быть как

устойчивый, так и неустойчивый. На массиве из n элементов имеет время выполнения в

худшем, среднем и лучшем случае Θ(n2), предполагая что сравнения делаются за

постоянное время

Это возможно, самый простой в реализации алгоритм сортировки. Как и в большинстве

других подобных алгоритмов, в его основе лежит операция сравнения. Сравнивая каждый

элемент с каждым, и в случае необходимости производя обмен, метод приводит

последовательность к необходимому упорядоченному виду.

Пусть имеется массив A размером N, тогда сортировка выбором сводится к следующему:

* берем первый элемент последовательности A[i], здесь i – номер элемента, для первого i равен 1;
* находим минимальный (максимальный) элемент последовательности и запоминаем его номер;
* если номер первого элемента и номер найденного элемента не совпадают, тогда два этих элемента обмениваются значениями, иначе никаких манипуляций не происходит;
* увеличиваем i на 1 и продолжаем сортировку оставшейся части массива.

С каждым последующим шагом размер подмассива, с которым работает алгоритм,

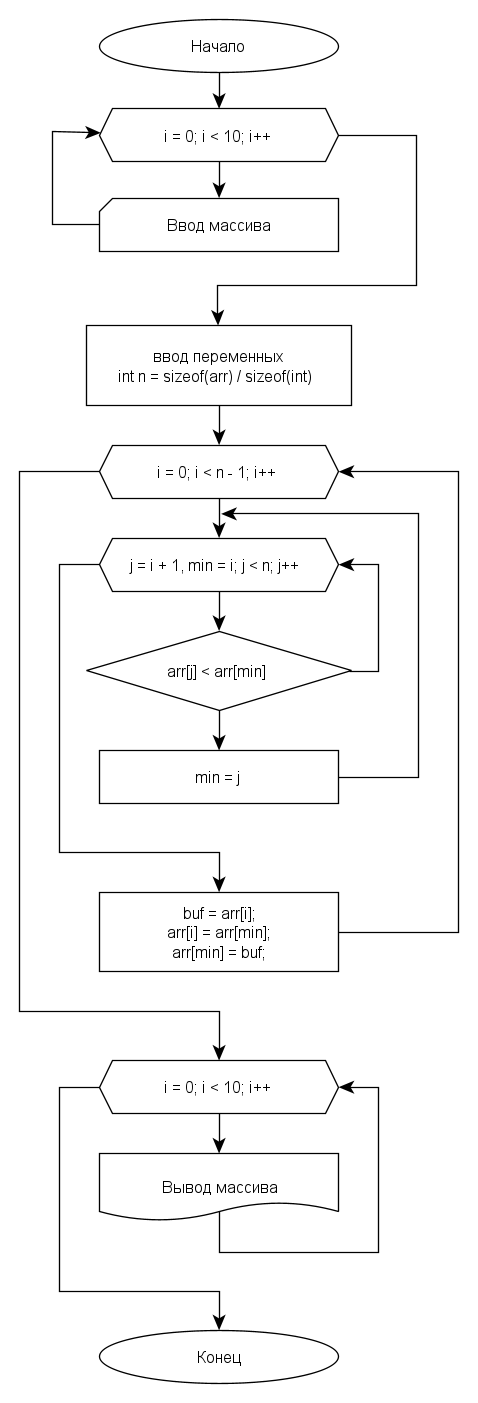
уменьшается.

Словесное представление алгоритма

Array – массив, n – длина массива, i,j – индексы массивов, min – индекс локального минимума

1. Сортировка начинается с первого элемента i=0
2. Если i<n-1, то п.3, иначе к пункту 12
3. min = i, j=i+1
4. Если j<n, то к п.5, иначе к пункту 9
5. Ищем локальный минимум. Если array[j] < array[min], то к пункту 6, иначе к пункту 7
6. Запоминаем новый индекс (min = j)
7. j++
8. К пункту 4
9. Обмен значениями array[i] и array[min]
10. i++
11. К пункту 2
12. Конец алгоритма

Блок-схема с использованием элемента модификации



Нет

Да

Рисунок 1 — Блок-схема 1 программы

Листинг программы с использованием параметрического цикла

Листинг 1 — Исходный код 1 программы

#include <iostream>

#include <stdlib.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian"); //установка русского языка

int i, j, min, buf; //ввод переменных

int arr[10]{}; // Объявляем массив из 10 элементов

printf("Введите 10 цифр в столбик:\n");

for (i = 0; i < 10; i++) // Вводим значения элементов массива через клавиатуру

{

scanf\_s("%d", &arr[i]);

}

int n = sizeof(arr) / sizeof(int); //размер массива

for (i = 0; i < n - 1; i++) //внешний цикл

{

for (j = i + 1, min = i; j < n; j++) //внутренний цикл, берем следующее число, пока первое не станет равным минимуму

{

if (arr[j] < arr[min]) //сраниваем 2 элемента

min = j; //устанавливаем начальное значение минимального элемента

}

buf = arr[i]; //меняем значения местами

arr[i] = arr[min];

arr[min] = buf;

}

printf("Отсортированный массив:\n");

for (i = 0; i < 10; i++) //вывод массива на экран

printf("%d ", arr[i]);

printf("\n");

return 0;

}

Результат работы 1 программы

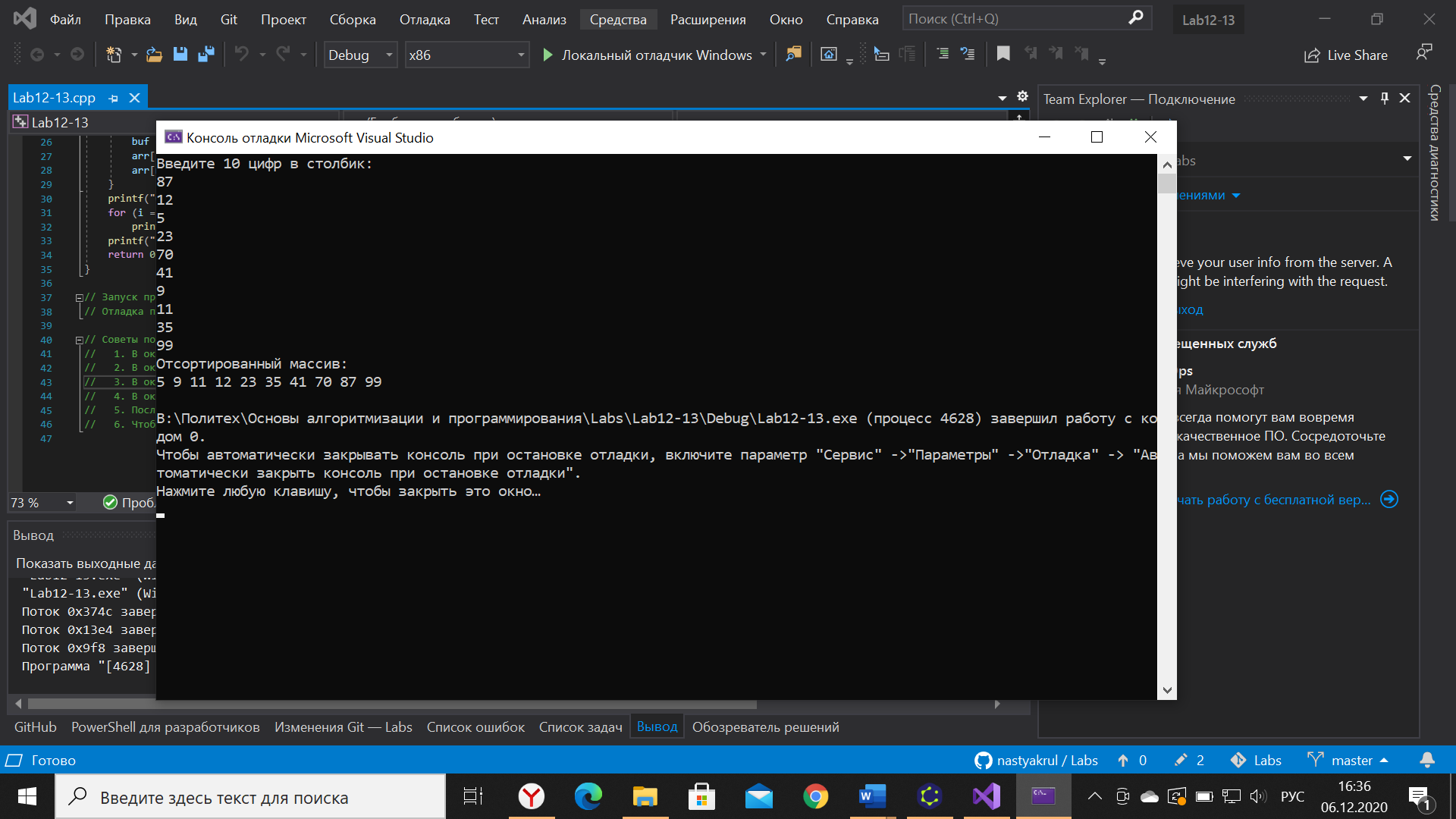
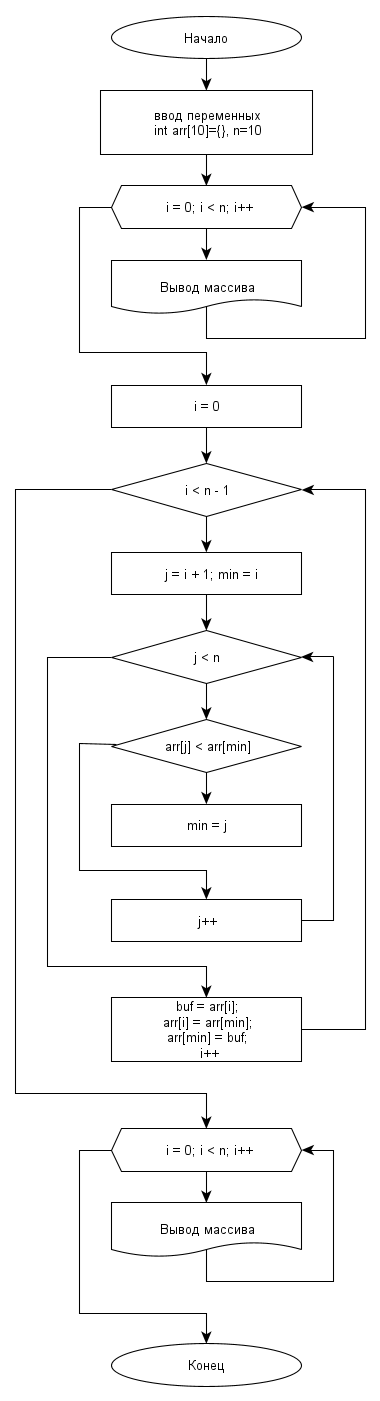


Рисунок 2 — Результат работы 1 программы

Блок-схема без использования элемента модификации



Нет

Нет

Нет

Да

Да

Да

Рисунок 3 — Блок-схема 2 программы

Листинг программы с использованием цикла с предусловием

Листинг 2 — Исходный код 2 программы

#include <iostream>

#include <stdlib.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian"); //установка русского языка

int i, j, min, buf, n = 10; //ввод переменных

int arr[10]={56, 12, 9, 88, 2, 74, 166, 35, 90, 1}; // Объявляем массив из 10 элементов

for (i = 0; i < n; i++) //вывод исходного массива на экран

printf("%d ", arr[i]);

printf("\n");

i = 0; //обнуляем

while (i < n - 1) //ввод внешнего цикла

{

j = i + 1; //присвоение значений

min = i;

while (j < n) //ввод внутреннего цикла

{

if (arr[j] < arr[min]) //поиск минимального числа

min = j; // изменение значения min

j++; //увеличение переменной

}

buf = arr[i]; //перестановка

arr[i] = arr[min];

arr[min] = buf;

i++; //увеличение переменной

}

printf("Отсортированный массив:\n");

for (i = 0; i < n; i++) //вывод преобразованного массива на экран

printf("%d ", arr[i]);

printf("\n");

return 0;

}

Результат работы 2 программы

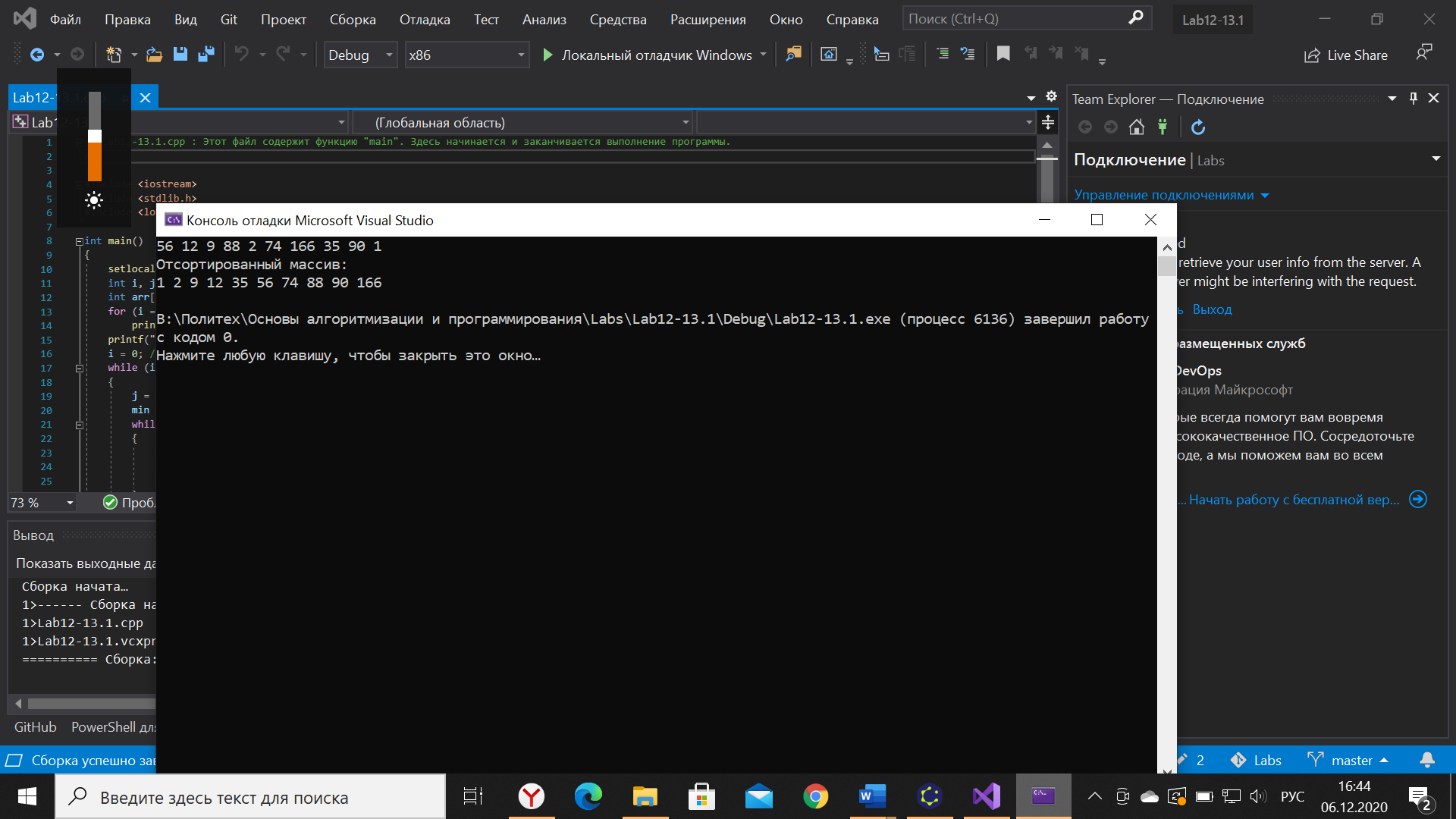


Рисунок 4 — Результат работы 2 программы