Министерство науки и высшего образования Российской Федерации

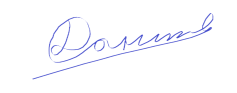
Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

ОСНОВЫ АЛГОРИТМИЗАЦИИ. СРЕДА VISUAL STUDIO. ВВЕДЕНИЕ В ЯЗЫК СИ.

Отчёт по лабораторной работе по дисциплине “Введение в программирование”

Студент гр. 543-1

\_\_\_\_\_\_\_\_\_ Д.А. Пляскин

13.09.2023.г

Доцент кафедры ЭМИС

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Е.А. Шельмина

оценка

“\_\_”\_\_\_\_\_\_ 2023 г.

Томск 2023

**Лабораторная работа №1**

**“Основы алгоритмизации. Среда Microsoft Visual Studio. Введение в язык Си”**

Введение: цель работы знакомство со средой разработки Visual Studio усвоение основ языка Си и Алгоритмизации.

Ход работы:

Основы Алгоритмизации: Алгоритмом называется точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи. В роли исполнителей алгоритмов могут выступать люди, роботы, компьютеры.

Блок схема – это графическое представление алгоритма при помощи стандартных обозначений. На схемах алгоритмов выполняемые действия изображаются в виде отдельных блоков, которые соединяются между собой линиями связи в порядке выполнения действий. Внутри блока дается информация о выполняемых действиях.

Таблица 1 – Основные блоки, используемые при составлении алгоритмов

|  |  |  |
| --- | --- | --- |
| **Название** | **Обозначение** | **Назначение** |
| Пуск, Остановка |  | Начало-конец алгоритма |
| Процесс |  | Любое вычислительное действие |
| Решение |  | Проверка условия |
| Ввод-вывод |  | Ввод-вывод данных |

Окончание таблицы 1

|  |  |  |
| --- | --- | --- |
| **Название** | **Обозначение** | **Назначение** |
| Соединитель |  | Используется на линиях разрыва |
| Комментарий |  | Комментарий |

Вычислительные процессы могут быть: линейные, разветвляющиеся и циклические.

Линейный алгоритм – алгоритм, в котором все команды выполняются последовательно друг за другом.

Часто при выполнении алгоритма должны предлагаться различные действия в зависимости от выполнения или невыполнения некоторого условия. Такие алгоритмические структуры называют ветвлением. Бывают алгоритмы с полным ветвлением (рисунок 1) и неполным (рисунок 2).

Рисунок 1 – Иллюстрация алгоритма с полным ветвлением

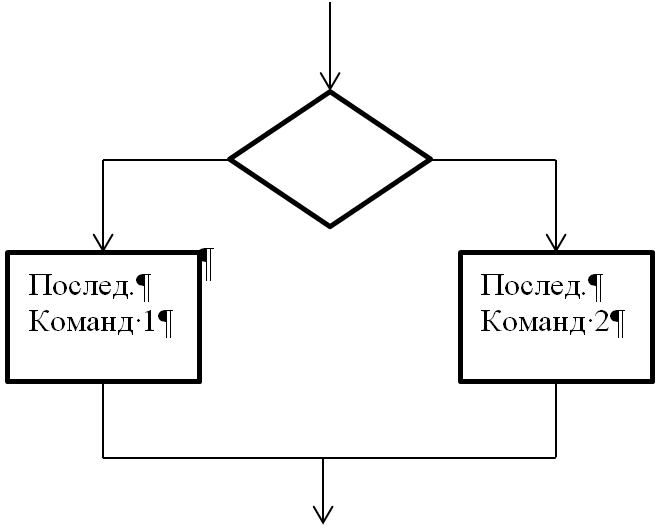
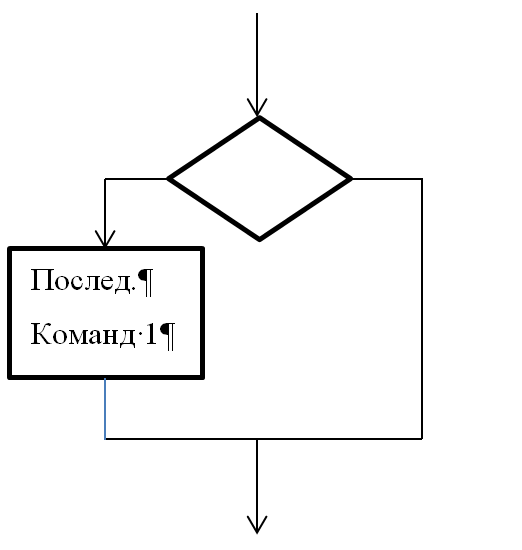


Рисунок 2 – Иллюстрация алгоритма с неполным ветвлением



Алгоритм с циклом – алгоритм, в котором присутствует многократное повторение одних и тех же действий. Такие алгоритмы подразделяются на алгоритмы с предусловием (рисунок 3) и постусловием (рисунок 4).

Рисунок 3 – Иллюстрация алгоритма с предусловным циклом

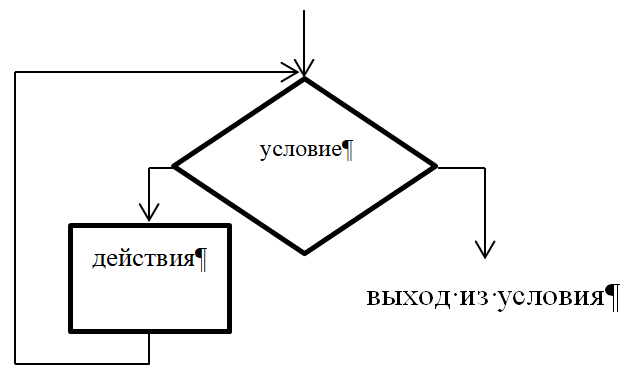
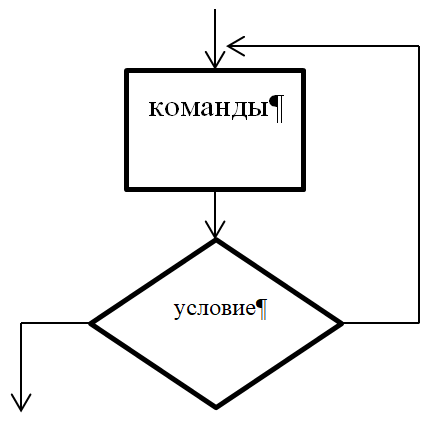
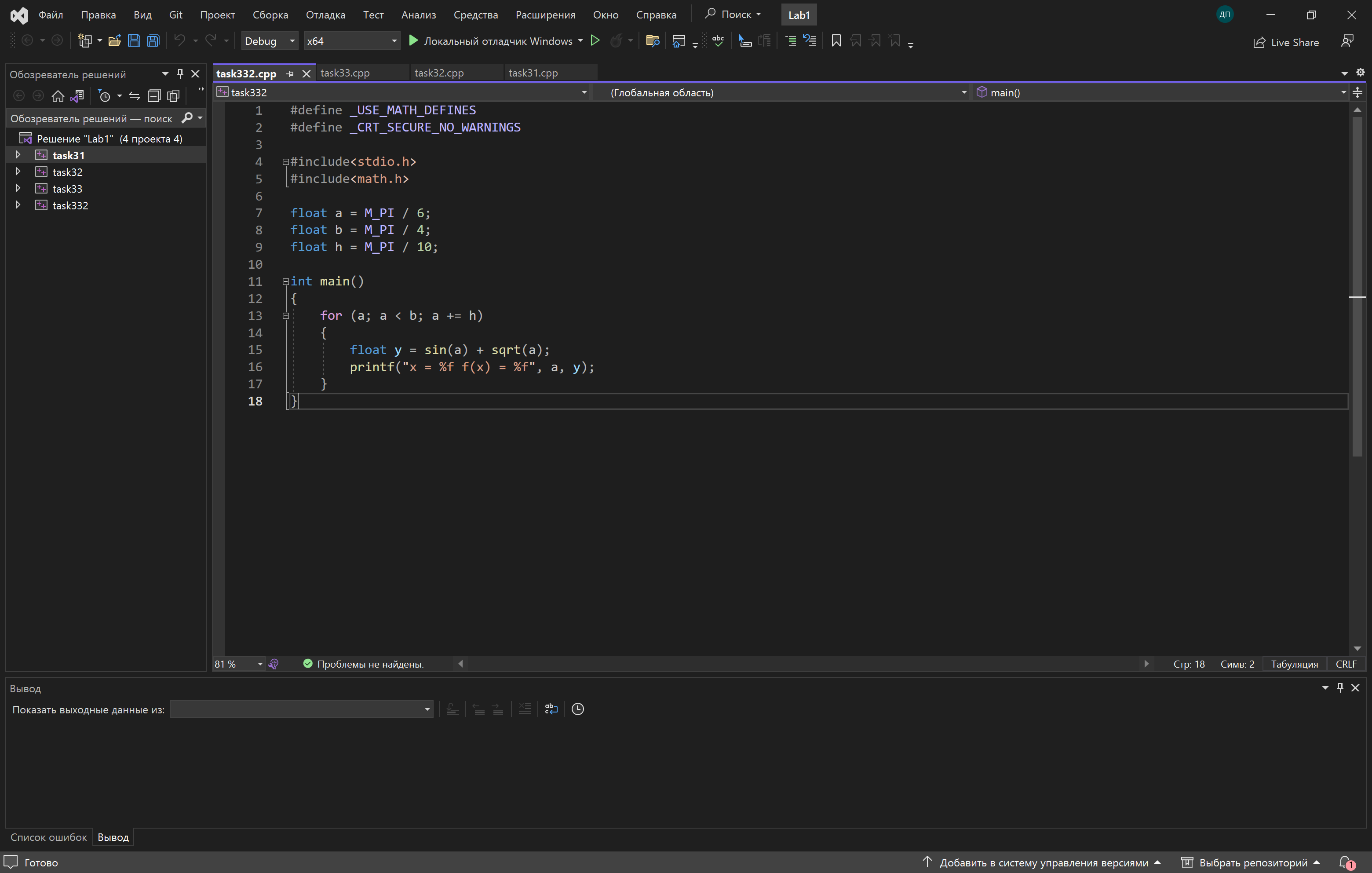


Рисунок 4 – Иллюстрация алгоритма с постусловным циклом



Среда программирования Microsoft Visual Studio: После открытия Microsoft Visual Studio (рисунок 5) можно увидеть три основные части интегрированной среды разработки: окна инструментов, меню с панелями инструментов и область главного окна. Окна 13 инструментов закреплены в левой и правой частях окна приложения, а панель Быстрый запуск, строка меню и стандартная панель инструментов закреплены в его верхней части. В центре окна приложения находится Начальная страница. При открытии решения или проекта редакторы и конструкторы отображаются в этом пространстве. При разработке приложения чаще всего используется именно эта область.

Рисунок 5 - Скриншот интегрированной среды разработки Microsoft Visual Studio



Создание простого приложения: При создании приложения в Microsoft Visual Studio необходимо сначала создать проект и решение.

Создание консольного приложения:

1. В строке меню выберите Файл, Создать, Проект.
2. В категории Visual C++ выберите шаблон Консольное приложение Win32 и назовите проект.
3. Когда появится мастер приложений Win32, нажмите кнопку Готово.

Проект и решение с основными файлами для консольного приложения Win32 создадутся и автоматически загрузятся в Обозреватель решений. Файл .cpp откроется в редакторе кода.

Добавление кода в приложение: Далее необходимо добавить код для отображения слова "Hello" в окне консоли. Для этого в окне консоли в файле .cpp введите пустую строку перед строкой return 0;, а затем введите в нее следующий код: printf("Hello\n");

Сохраните изменения в файле.

Отладка и тестирование приложения: С помощью отладки можно посмотреть, отображается ли слово Hello в окне консоли.

Отладчик запустится и выполнит код. Окно консоли (отдельное окно, подобное командной строке) отображается в течение нескольких секунд, но при остановке отладчика быстро закрывается. Чтобы просмотреть текст, необходимо установить точку останова выполнения программы.

Добавление точки останова:

1. Добавьте точку останова из меню в строке return 0;. Для установки точки останова можно также просто щелкнуть область слева. Рядом со строкой кода в крайнем левом поле окна редактора появится красный кружок

2. Нажмите клавишу F5, чтобы начать отладку. Запускается отладчик, и появляется окно консоли, в котором выводится слово Hello.

3. Для останова процесса отладки нажмите SHIFT + F5.

Сборка окончательной версии приложения: Теперь, когда вы проверили, что все работает, можно подготовить окончательную сборку приложения.

Очистка файлов решения и сборка окончательной версии

1. Из строки меню удалите промежуточные файлы и выходные файлы, созданные во время предыдущих сборок.

2. Измените конфигурацию сборки с Отладка на Выпуск.

3. Постройте решение

Структура программы на Си. Функция ввода-вывода: Рассмотрим простую программу на С — печать строки текста:

/\* Первая программа на С \*/

#include<stdio.h>

main()

{

printf("Welcome to C!\n");

}

Результат работы программы:

Welcome to С!

Начинается символами /\* и заканчивается символами \*/, означающими, что эта строка является комментарием. Программисты вставляют в код комментарии для документирования программ и для того, чтобы сделать их более удобочитаемыми. Комментарии не оказывают никакого влияния на работу компьютера во время исполнения программы. Комментарий Первая программа на С просто объясняет назначение программы.

Строка:

main()

Должна обязательно присутствовать в каждой программе. Скобки после main означают, что main является «строительным блоком» программы, называемым функцией. Программа С может содержать одну или большее количество функций, однако одна из функций обязательно должна быть main.

Левая фигурная скобка ({) должна предварять тело каждой функции. Соответственно правая фигурная скобка должна стоять в конце каждой функции. Эта пара скобок и часть программы между ними называется блоком

Строка:

printf("Welcome to C!\n");

Дает компьютеру команду выполнить действие, а именно вывести на экран строку символов, находящуюся внутри кавычек. Такую строку называют символьной строкой, сообщением или литералом. Вся строка, включая printf, аргументы внутри круглых скобок и точку с запятой (;), называется оператором. Каждый оператор должен заканчиваться точкой с запятой (иногда называемой символом конца оператора). Результатом выполнения оператора printf является вывод сообщения Welcome to С! на экран. Символы обычно печатаются именно так, как они записаны внутри двойных кавычек в операторе printf. Заметьте, что символы \n не появились на экране. Обратная косая черта (\) называется esc-символом. Он указывает, что printf предстоит выполнить нечто нестандартное. Когда встречается обратная косая черта, printf считывает следующий за ним символ и, объединяя его с обратной косой четой, создает esc-код. Esc-код \n означает новую строку, результатом является перевод курсора на начало следующей строки на экране. Функция printf — одна из многих функций, входящих в стандартную библиотеку.

Правая фигурная скобка (}) означает, что выполнение функции main окончено.

Задания для самостоятельной работы:

Задание 1.1. Составить блок-схему решения линейного алгоритма:

Для выполнения задания 1 (рисунок 6) использовался вариант 16, представленный в файле лабораторной работы №1.

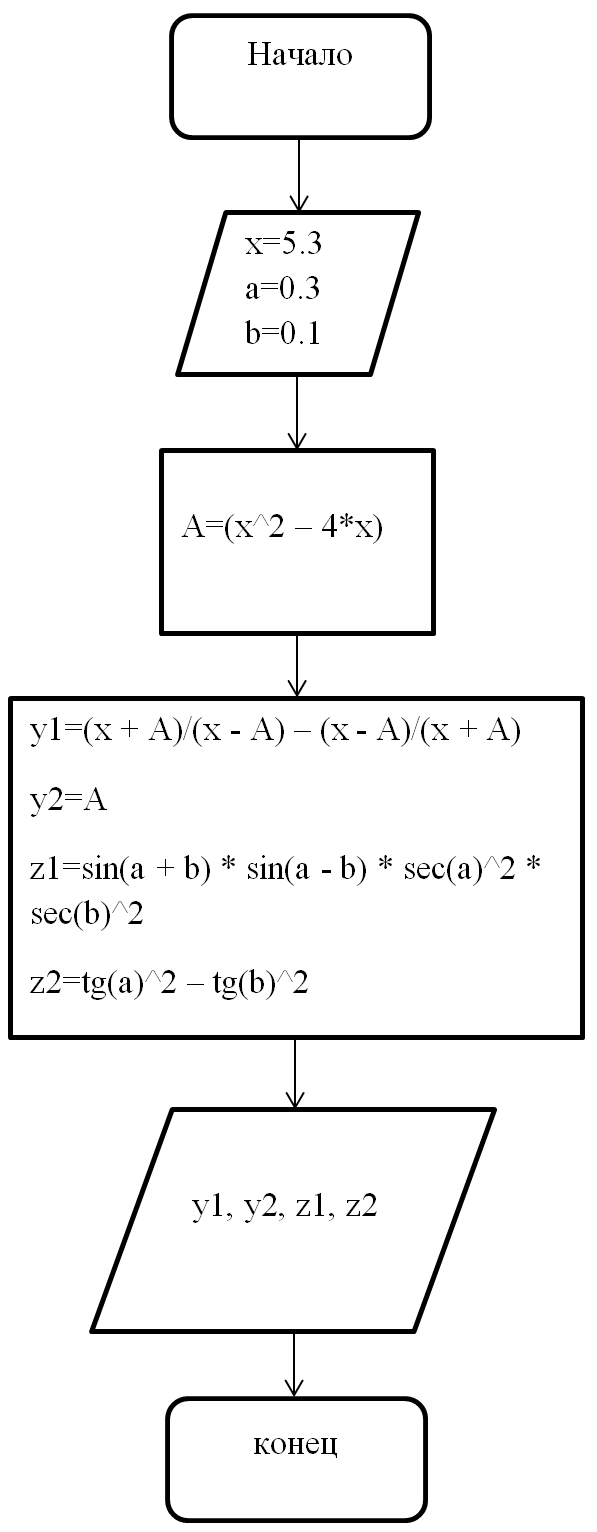
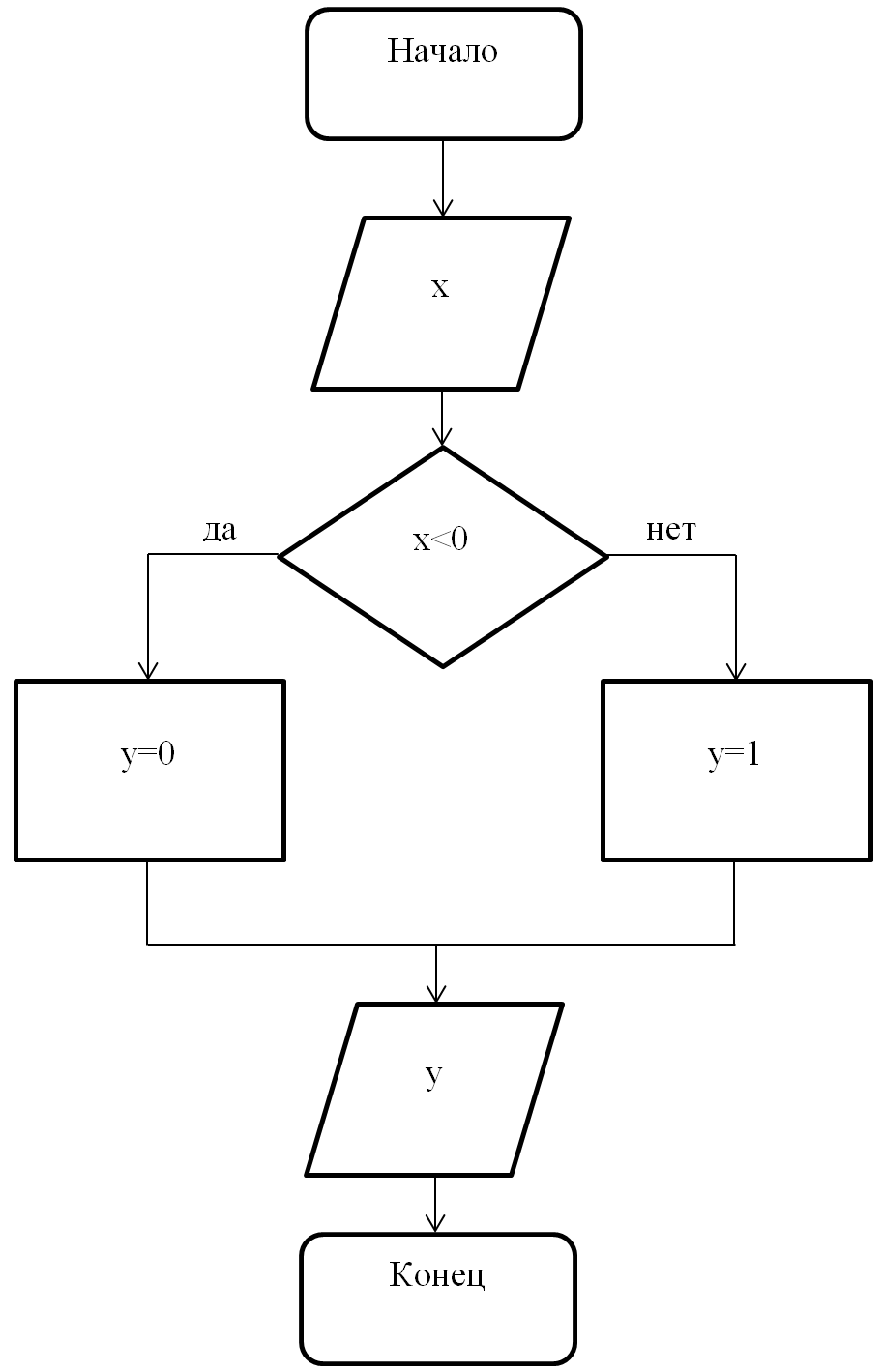


Рисунок 6 – Иллюстрация блок-схемы задания 1

Задание 1.2. Составить блок-схему решения разветвляющегося алгоритма:

Для выполнения задания 2 (рисунок 7) использовался вариант 16, представленный в файле лабораторной работы №1.

Рисунок 7 – Иллюстрация блок-схемы задания 2



Задание 1.3. Составить блок-схему решения циклического алгоритма.

Для выполнения задания 3 (рисунок 8) использовался вариант 16, представленный в файле лабораторной работы №1.

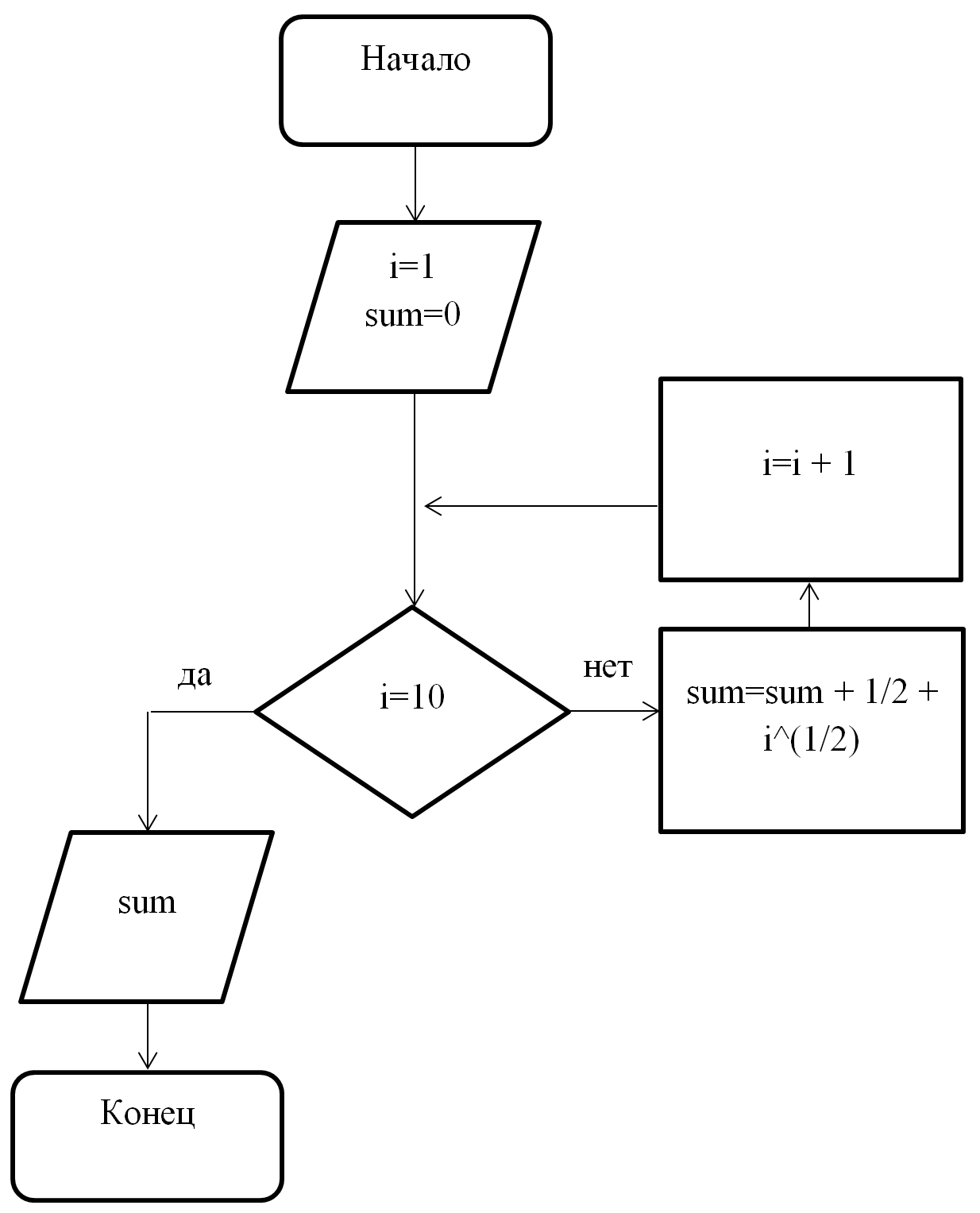
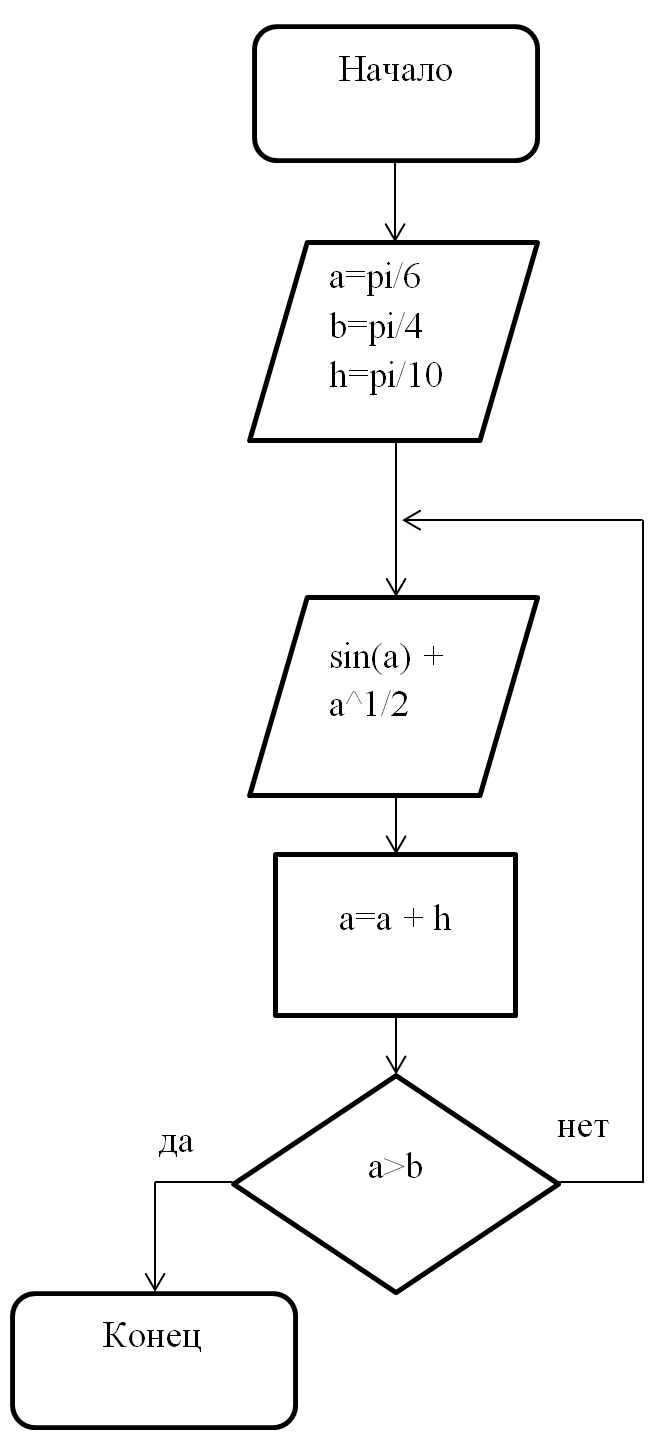


Рисунок 8 – Иллюстрация блок-схемы задания 3

Задание 1.3.2. Составить блок-схему алгоритма вычисления значений функции f(x) на отрезке [a;b] с шагом h.

Для выполнения задания 3.2 (рисунок 9) использовался вариант 16, представленный в файле лабораторной работы №1.

Рисунок 9 – Иллюстрация блок-схемы задания 3.2



Задание 3.1. Выполнить примеры 1-3.

Для выполнения примеров, показанных на рисунках 10, 11 и 12, использовалась среда разработки Microsoft Visual Studioи шаблон Консольное приложение.

Рисунок 10 – Скриншот кода из примера 1

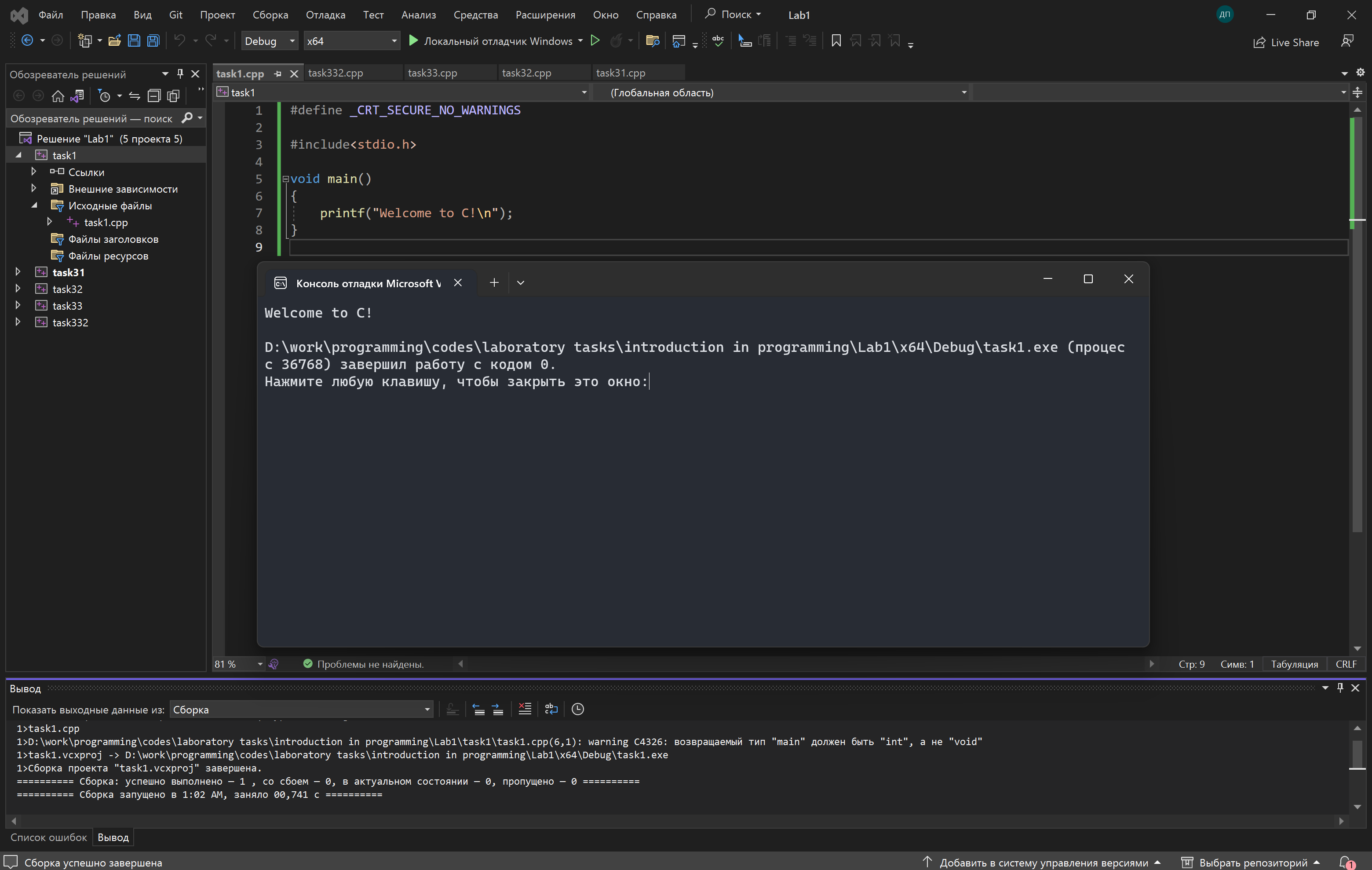


Рисунок 11 – Скриншот кода из примера 2

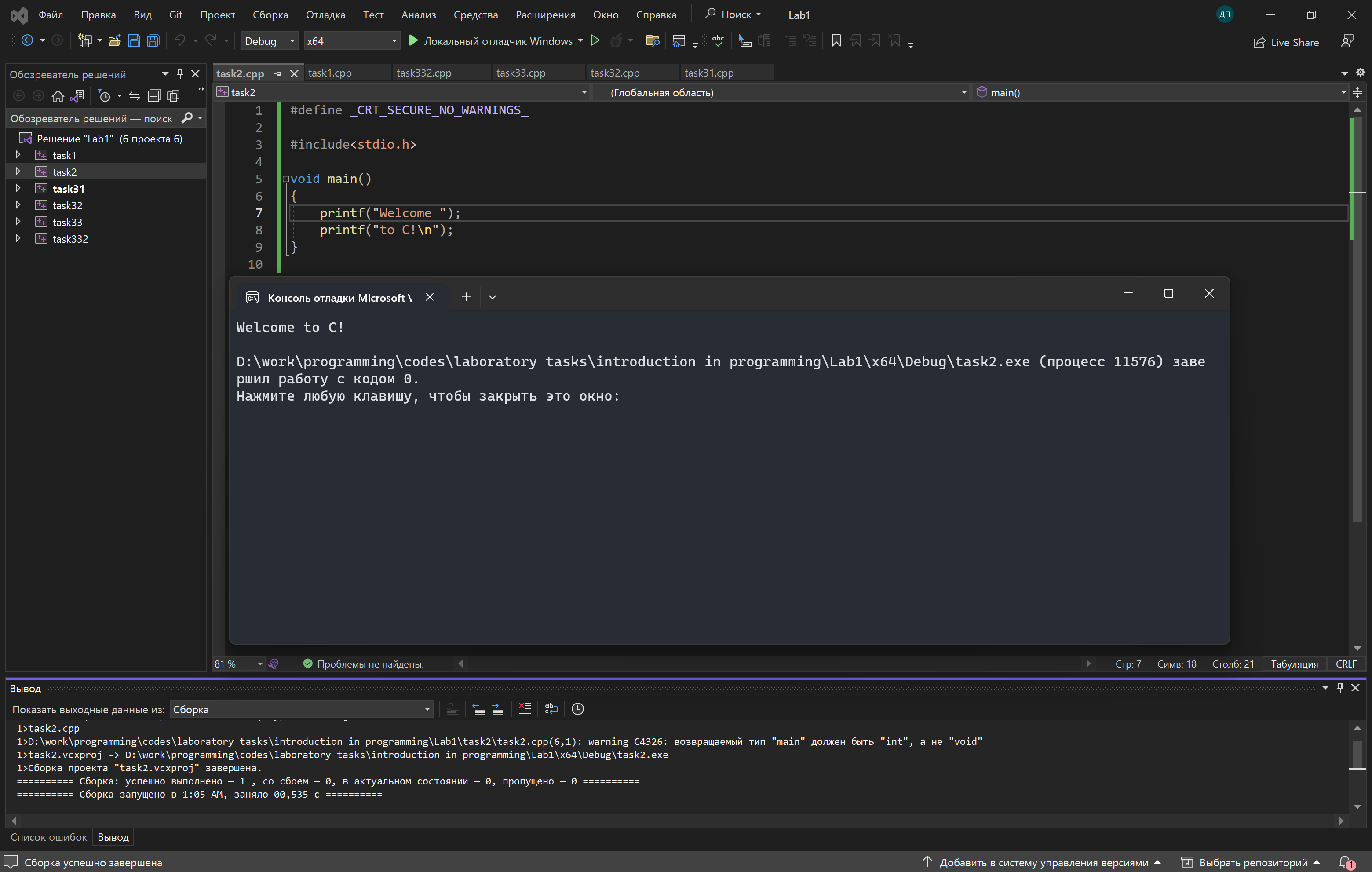
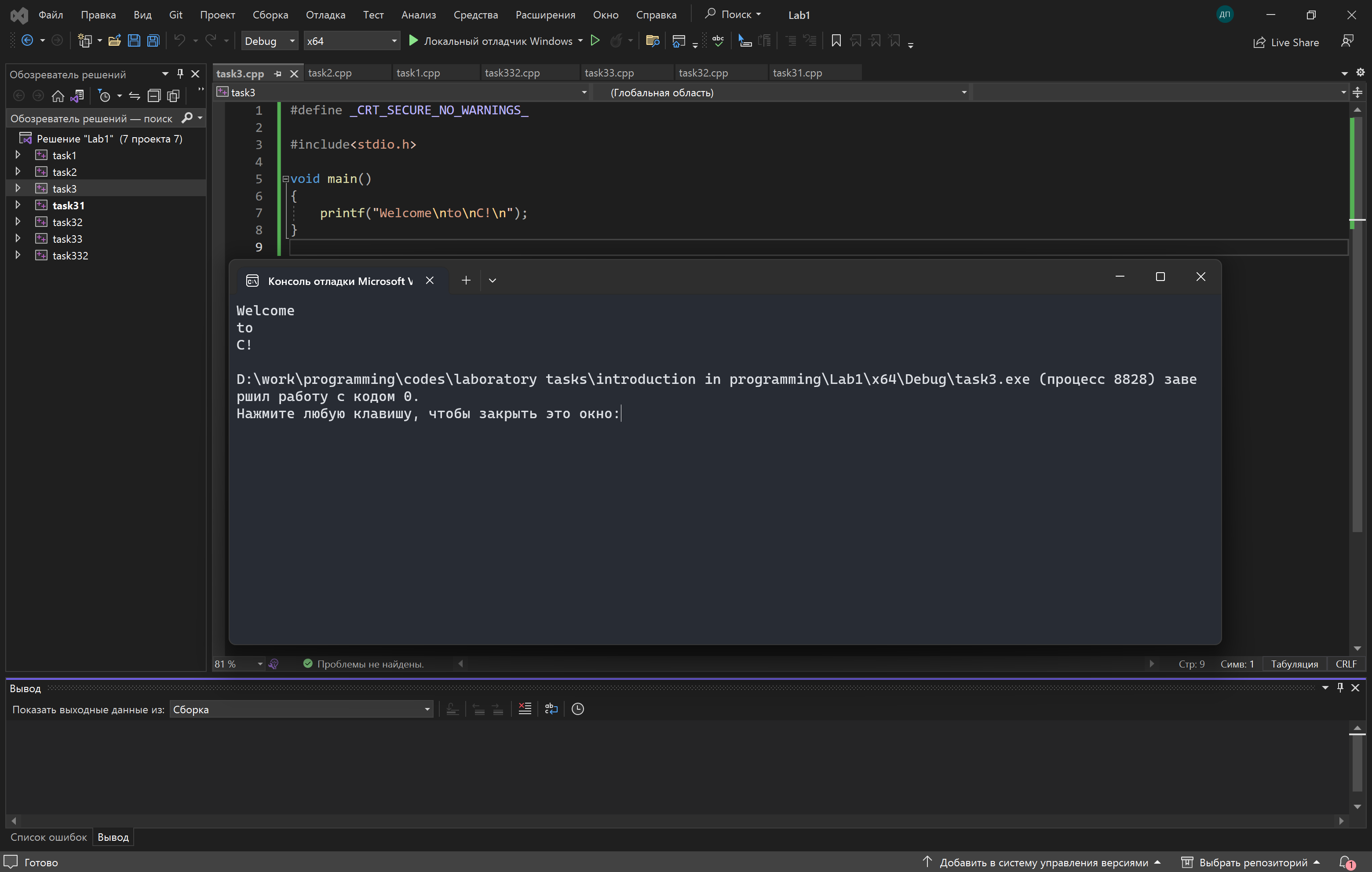


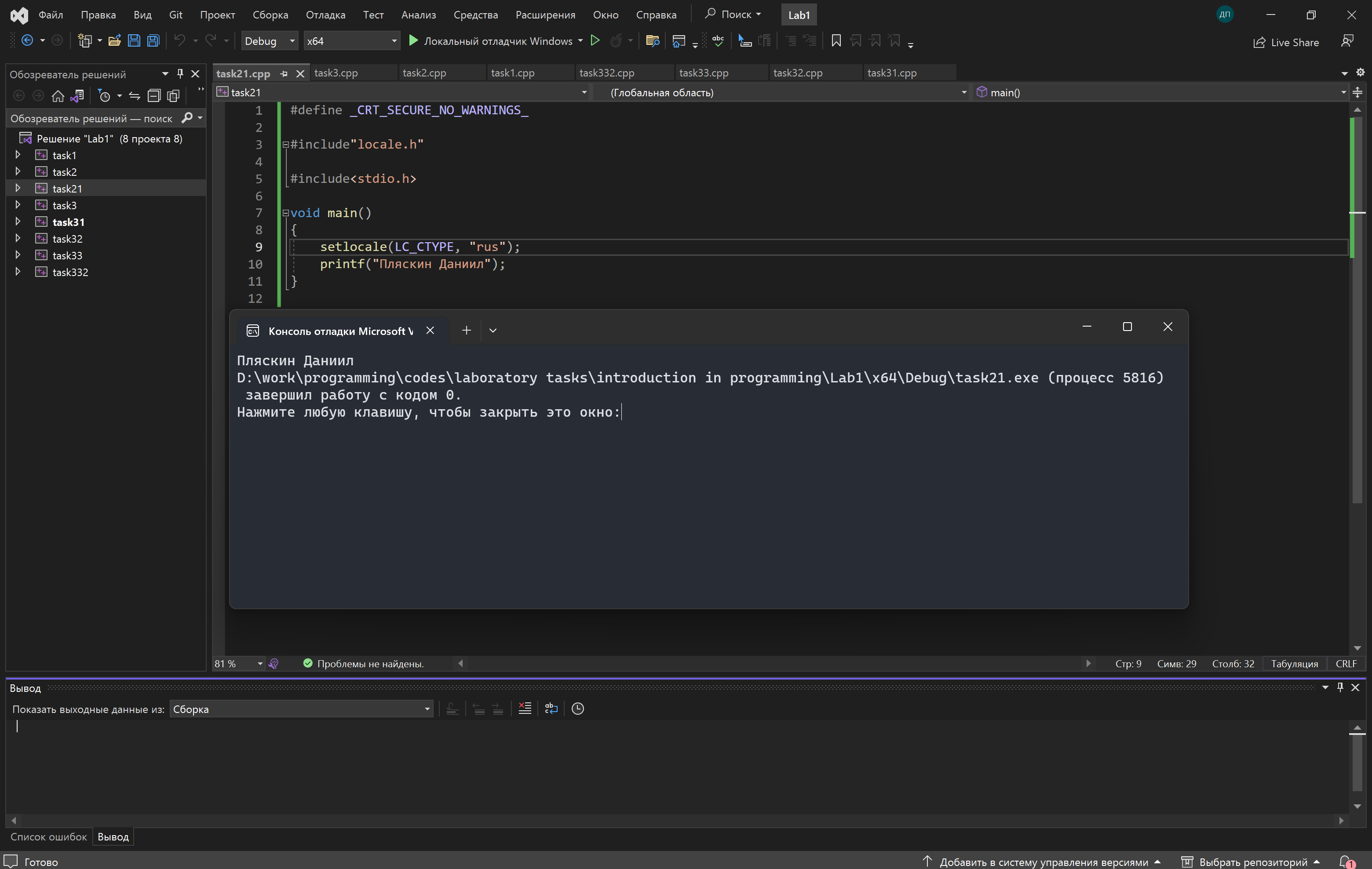
Рисунок 12 – Скриншот кода из примера 3



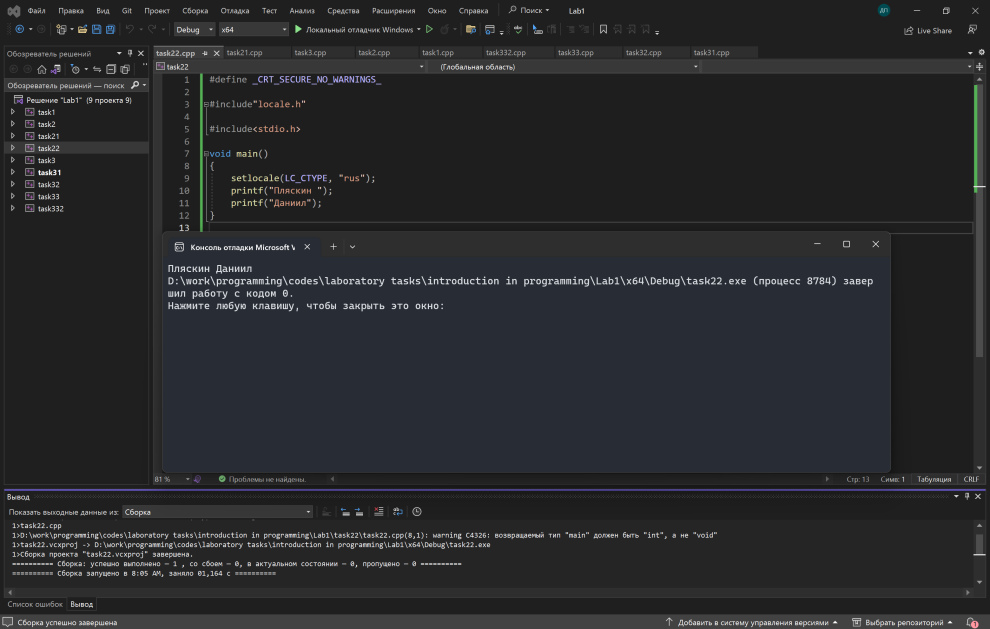
Задание 3.2: Напишите программу, выводящую на печать вашу фамилию и имя тремя способами.

Для выполнения задания (рисунок 13) в одну строку, применив один оператор printf, использовалась среда разработки Microsoft Visual Studio и шаблон Консольное приложение.

Рисунок 13 – скриншот выполненного задания первым способом

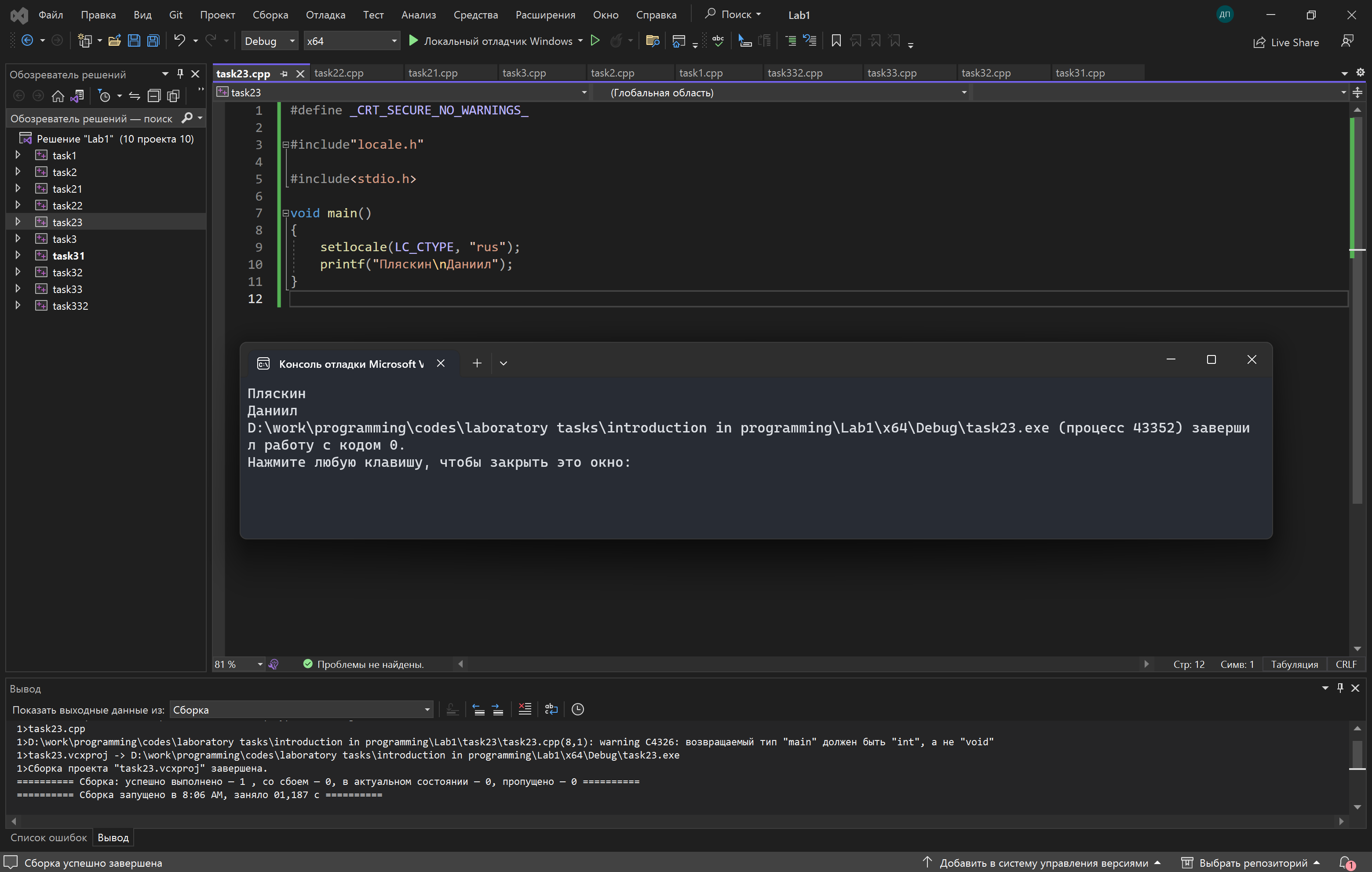


Для выполнения задания (рисунок 14), в одну строку, применив два оператора printf, использовалась среда разработки Microsoft Visual Studio и шаблон Консольное приложение.

Рисунок 14 – Скриншот выполненного задания вторым способом

Для выполнения задания (рисунок 15), в одну строку, применив два оператора printf, использовалась среда разработки Microsoft Visual Studio и шаблон Консольное приложение.

Рисунок 15 – Скриншот выполненного задания третьим способом



Вывод: В результате выполнения лабораторной работы были получены навыки создания блок-схем и работы с интерфейсом Microsoft Visual Studio.