МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 1**

**по учебной дисциплине «Основы алгоритмизации и программирования»**

**Тема: «Программирование алгоритмов линейной структуры»**

Выполнила студентка

специальности 09.02.07

Информационные системы и программирование

II курса группы 22919/9

Кнороз Варвара

Сергеевна

Преподаватель

Молькова Лолита Юрьевна

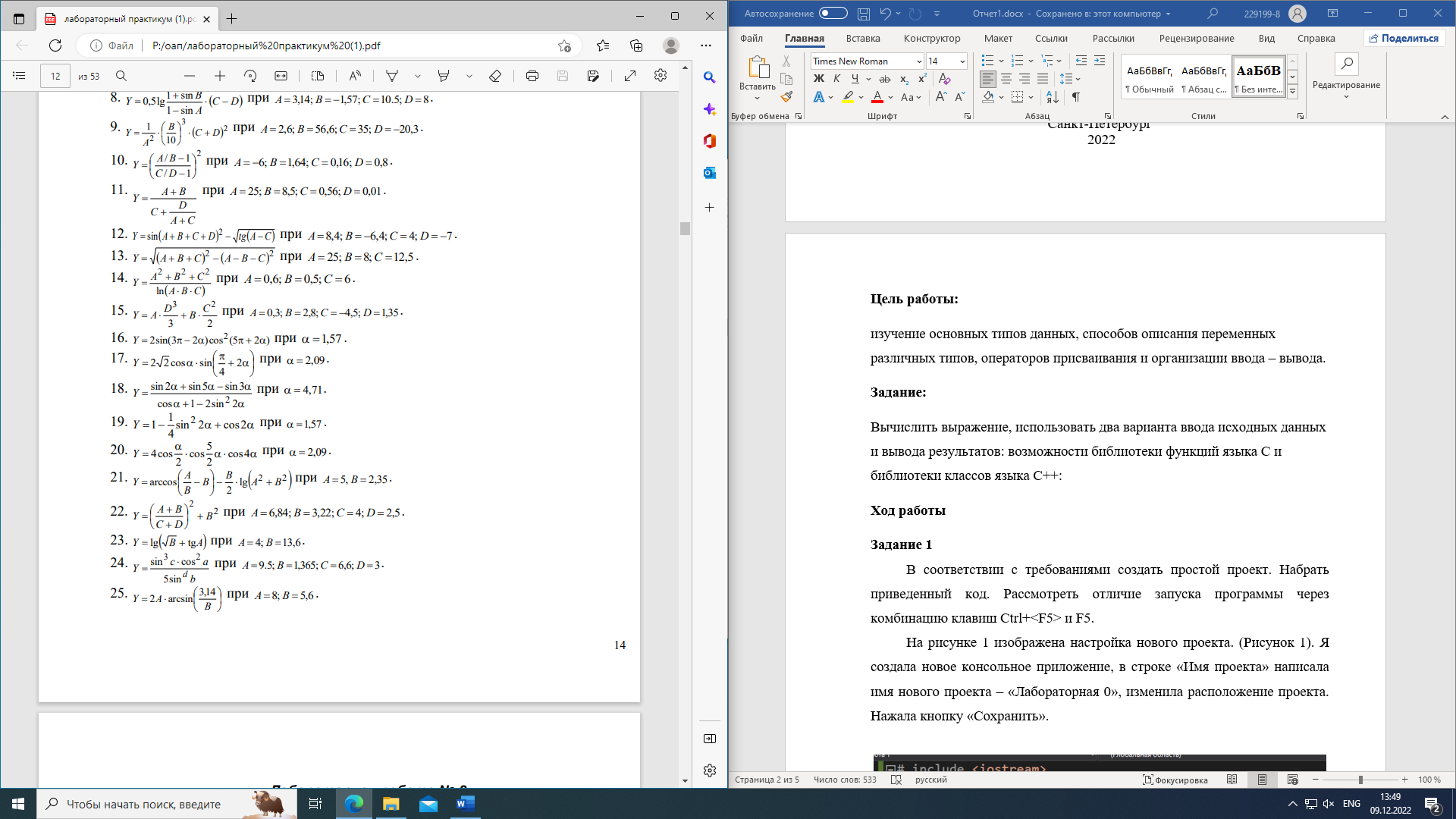
Санкт-Петербург

2022

Цель работы:

Изучение основных типов данных, способов описания переменных различных типов, операторов присваивания и организации ввода – вывода.

Задание:

Вычислить выражение

и использовать два варианта ввода исходных данных и вывода результатов: возможности библиотеки функций языка С и библиотеки классов языка С++:

Ход работы

Схема программы для задания представлена на рисунке 1:

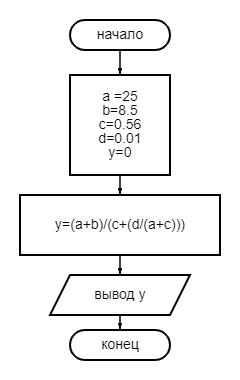


Рисунок 1 – схема работы программы

Текст программы с использованием библиотеки функций языка С (форматированный ввод-вывод) для ввода исходных данных и вывода результатов представлен на рисунке 2:

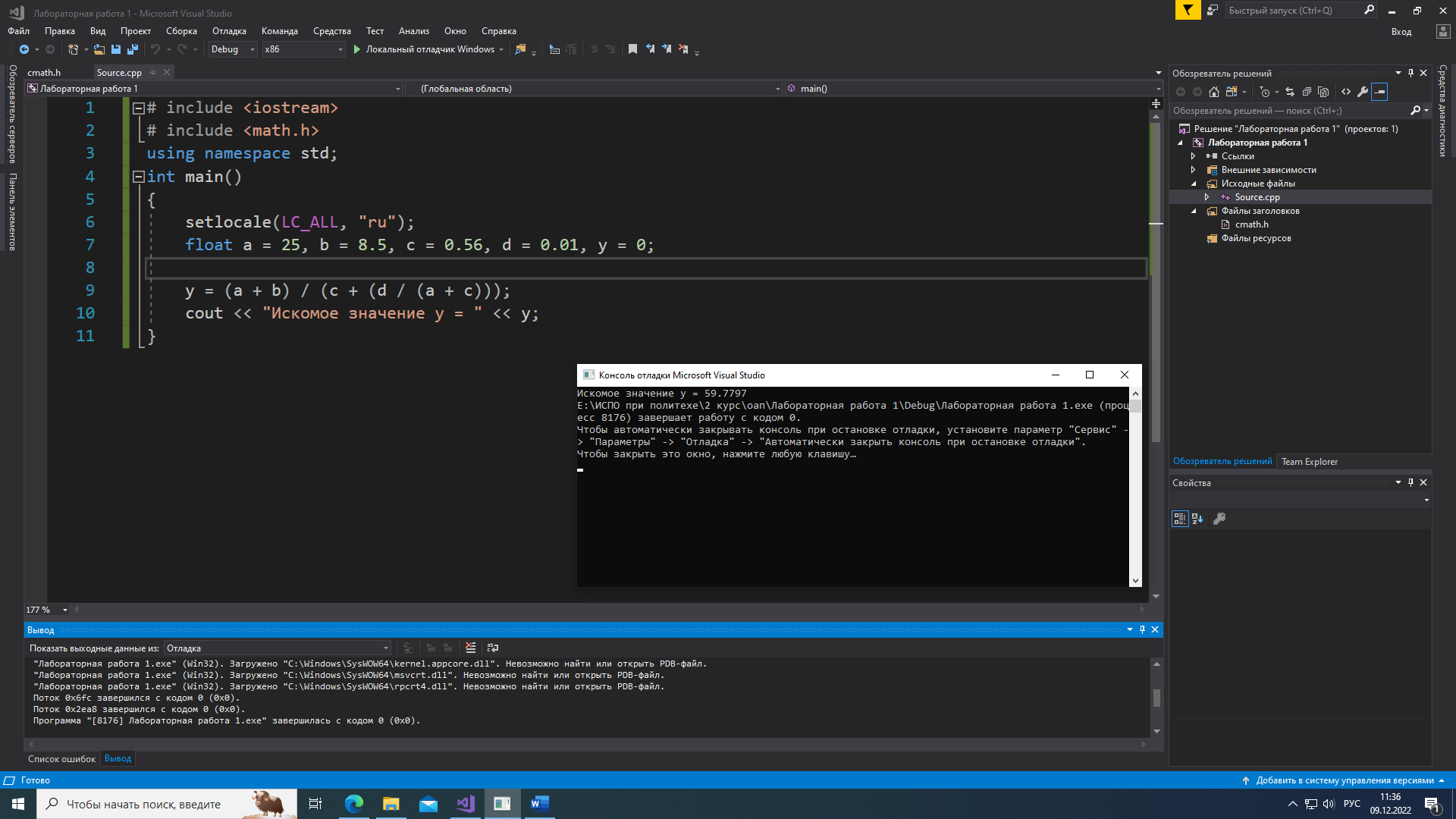


Рисунок 2 – код программы

Итог работы программы представлен на рисунке 3:

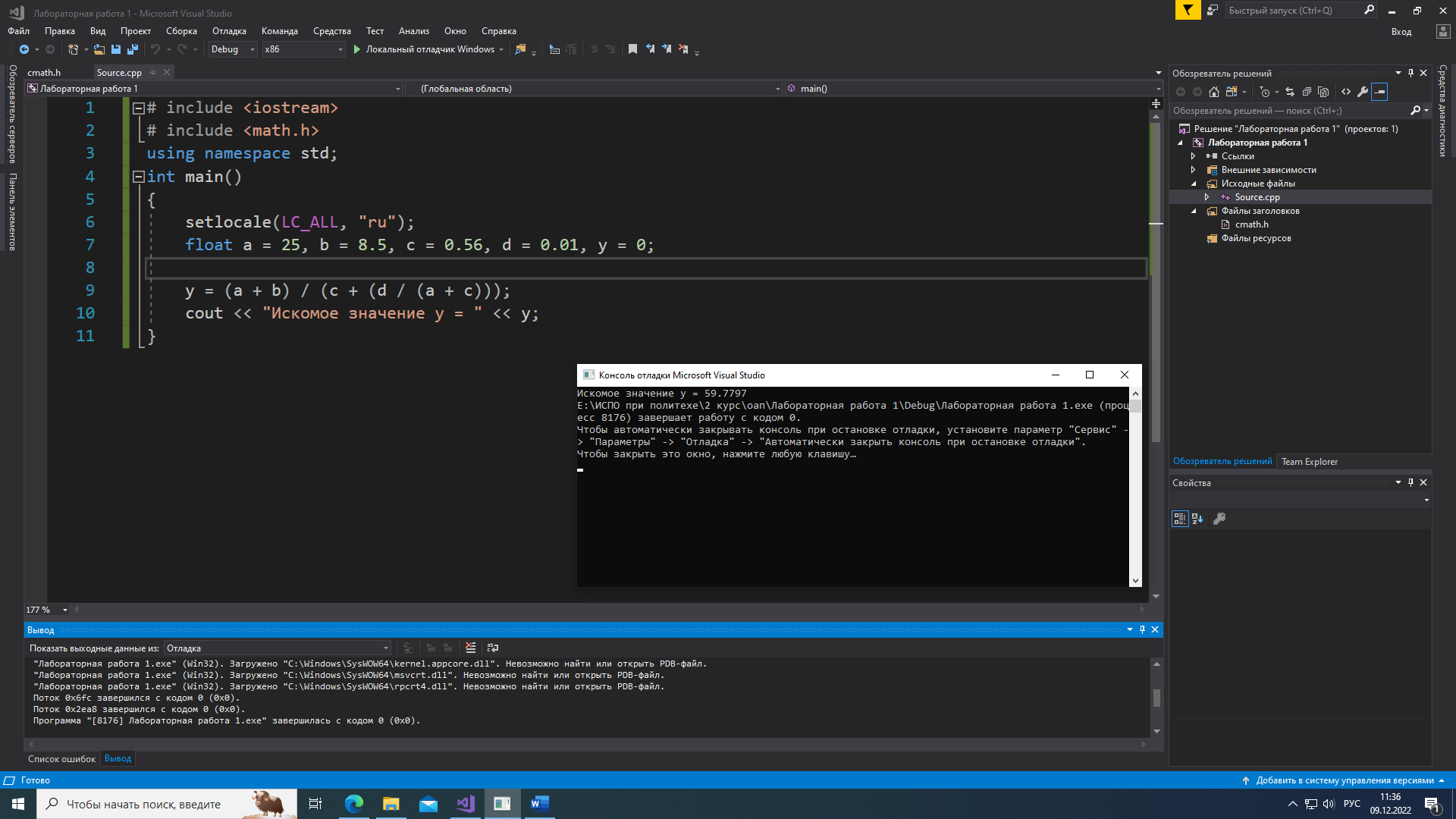


Рисунок 3 – итог работы программы

Текст программы с использованием библиотеки классов языка С++ (потоковый ввод-вывод) для ввода исходных данных и вывода результатов представлен на рисунке 4:

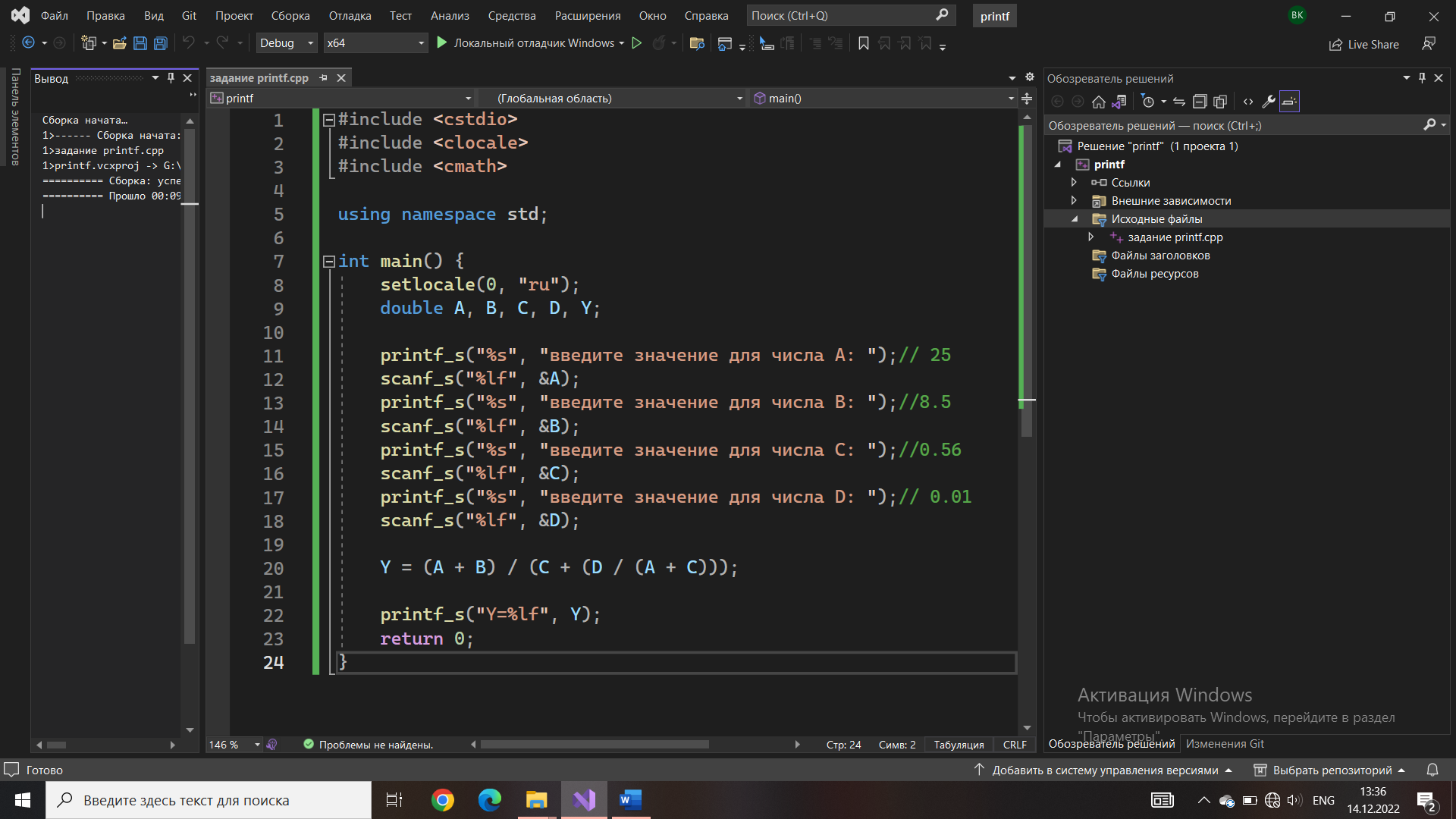


Рисунок 4 – код программы

Итог работы программы представлен на рисунке 5:

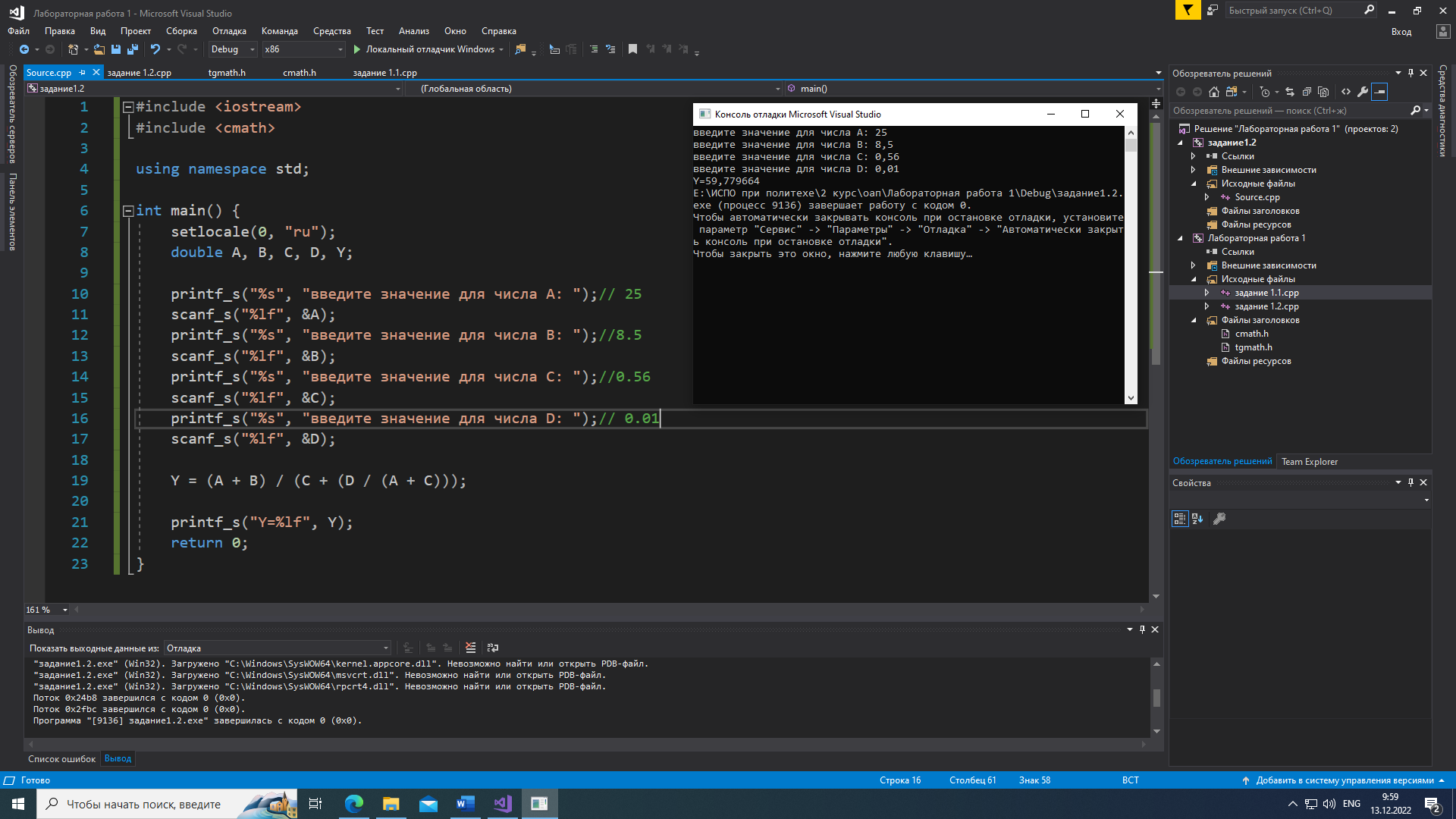


Рисунок 5 – итог работы программы

**Контрольные вопросы:**

1. Программа состоит из одного или нескольких файлов, образующих проект, файлы содержат описания данных и функций (обособленный участок программы, решающий часть задачи). Главная функция (main)– точка входа в программу, представляет решение всей задачи (обычно с использованием других функций). Для записи содержания действий, выполняемых функцией, используются управляющие конструкции языка программирования.
2. Директивы препроцессора, заголовочные файлы, прототипы библиотечных функций, их вызовы. #include — вставляет текст из указанного файла  
   #define — задаёт макроопределение (макрос) или символическую константу  
   #undef — отменяет предыдущее определение  
   #if — осуществляет условную компиляцию при истинности константного выражения  
   #ifdef — осуществляет условную компиляцию при определённости символической константы  
   #ifndef — осуществляет условную компиляцию при неопределённости символической константы  
   #else — ветка условной компиляции при ложности выражения  
   #elif — ветка условной компиляции, образуемая слиянием else и if  
   #endif — конец ветки условной компиляции  
   #line — препроцессор изменяет номер текущей строки и имя компилируемого файла  
   #error — выдача диагностического сообщения  
   #pragma — действие, зависящее от конкретной реализации компилятора. Заголовочный файл (файл заголовка, подключаемый файл или header file). Они имеют расширение .h, но иногда их можно увидеть и с расширением .hpp или вообще без расширения. Целью заголовочных файлов является удобное хранение предварительных объявлений для использования другими файлами. Всё содержимое из заголовочного файла копируется в файл \*.срр, т.е. всё содержимое становится доступным для использования. Заголовочные файлы: — cstdio (stdio.h)-заголовочный файл для выполнения операций ввода/вывода; — cstring (string.h)-заголовочный файл для работы со строками; — iostream-заголовочный файл с классами, функциями и переменными для организации ввода-вывода. Для удобства в библиотеке определены три стандартных объекта-потока: cin – объект класса istream, соответствующий стандартному вводу. В общем случае он позволяет читать данные с терминала пользователя; cout – объект класса ostream, соответствующий стандартному выводу. В общем случае он позволяет выводить данные на терминал пользователя; cerr – объект класса ostream, соответствующий стандартному выводу для ошибок. В этот поток мы направляем сообщения об ошибках программы. Вывод осуществляется, как правило, с помощью перегруженного оператора сдвига влево (<>).
3. Этапы обработки текста программы. Включение текстов из заголовочных файлов.

Компиляция программы на языке Си состоит из двух этапов: начальной обработки, выполняемой т.н. препроцессором, и собственно компиляции. Результат обработки исходного модуля компилятором - объектный код (object code, обычно хранится в файлах с расширением \*.obj). В большинстве случаев компилятор переводит на машинный язык каждую функцию по отдельности, не связывая их друг с другом и с библиотечными функциями; поэтому объектный код не может еще быть выполнен.

Лишь вторая специальная программа - редактор связей (Linker) - создает исполняемый (абсолютный, загрузочный) код, "связывая" между собой функции и места их использования (вызова), подключая при этом и библиотечные функции. Linker еще называют компоновщиком, т.к. он объединяет программные модули в новый модуль, пригодный для выполнения - загрузочный модуль (Load module, расширение \*.exe).

Большинство сред программирования по команде "Запуск программы" ("Run") автоматически выполняют все указанные этапы. Однако знать их различие все же стоит, чтобы при возникновении ошибок лучше понимать их причины. По тексту сообщения об ошибке обычно четко видно, на каком этапе она произошла.

1. Главная функция программы. Структура функции, ее заголовок.

Функция состоит из заголовка и тела. Заголовок функции – это ее интерфейсная (описательная) часть, которая не дает программного кода, а содержит описание входных и выходных параметров, необходимых при ее выполнении (вызове). Тело функции – это программный код (блок), ограниченный фигурными скобками. Тело функции при трансляции преобразуется во внутреннее (двоичное) представление и размещается в сегменте команд программы. Существует также синтаксис, в котором присутствует только заголовок, ограниченный символом «;». В этом случае речь идет об объявлении функции – информировании транслятора о ее наличии и интерфейсе.

В заголовке находится имя функции, после которого в скобках – список формальных параметров, разделенных запятыми. Список может быть и пустым. Синтаксис формального параметра – это синтаксис определения переменной, что наводит на мысль, что формальные параметры – это особого рода переменные, которые используются для передачи параметров (значений) при вызове (выполнении) функции.

Внутри тела функции могут находиться определения обычных переменных, которые называются локальными (в терминологии Си - автоматическими). Тело каждой функции представляет собой отдельное изолированное пространство, в котором допустимо использование собственных переменных, составляющих неявное окружение – контекст функции. В него входят вышеуказанные формальные параметры и локальные переменные. Кроме того, контекст функции является короткоживущим: он создается при входе в функцию и разрушается при завершении ее выполнения.

Формальные параметры используются для передачи входных и выходных данных в функцию. Кроме них функция имеет еще одну неявную выходную переменную – результат функции. Результат функции — это временная переменная, которая возвращается функцией и может быть использована как операнд в той части выражения, где был произведен ее вызов.

1. Определение переменных в программе. Типы переменных.

Среднее значение - av, количество элементов в последовательности - nn, индекс начала последовательности – ib. Для большинства данных достаточно типа short или char, т.к. их значения укладываются в диапазон: -128 - 128. Мы выбираем тип int согласно с общим стилем программирования на языке C.

1. Функции форматного ввода-вывода в стиле С.

Для использования данных функций необходимо подключение стандартной библиотеки stdio.h

printf() — для вывода информации

scanf() — для ввода информации.

1. Ввод-вывод данных потоком в стиле С++.

Для использования данных функций необходимо подключение стандартной библиотеки iostream

Cout — для вывода информации

Cin — для ввода информации.

1. Операции, выражения. Оператор присваивания.

Операция присваивания в языке программирования C++ обозначается знаком «=». Как и другие операторы в C++, она может быть перегружена. Операция присваивания копированием - особый вид операции присваивания, используемый для присваивания объектов одного класса друг другу. Является одним из особых членов-функций и генерируется автоматически компилятором в случае, если нет явного объявления программистом.