СОДЕРЖАНИЕ

[Практическая работа №1 «Основы программирования на с/с++» 2](#_Toc146274054)

[Практическая работа №2 «Циклы и ветвления» 8](#_Toc146274055)

[Практическая работа №3 «Структуры и перечисления» 16](#_Toc146274056)

[Практическая работа №4 «Функции» 23](#_Toc146274057)

Практическая работа №1 «Основы программирования на с/с++»

Цель работы: получить практические навыки в использовании в языках С и С++ для создания программ.

Задание №1: используя средства языка С и С++ (printf(), cout) напишите программу, выводящую па экран ваше любимое стихотворение. Для разбиения на строчки используйте подходящую управляющую последовательность.

Решение:

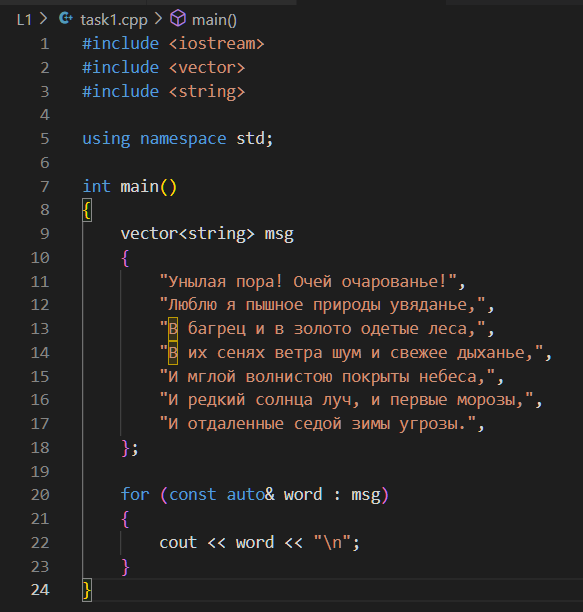


Рисунок – L1 Task1

Результат работы программы:

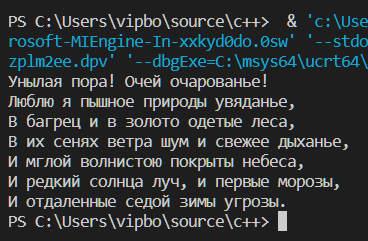


Рисунок - результат L1 Task1

Задание №2: напишите программу, запрашивающую температуру в градусах по Цельсию и отображающую ее эквивалент по Фаренгейту.

Решение:

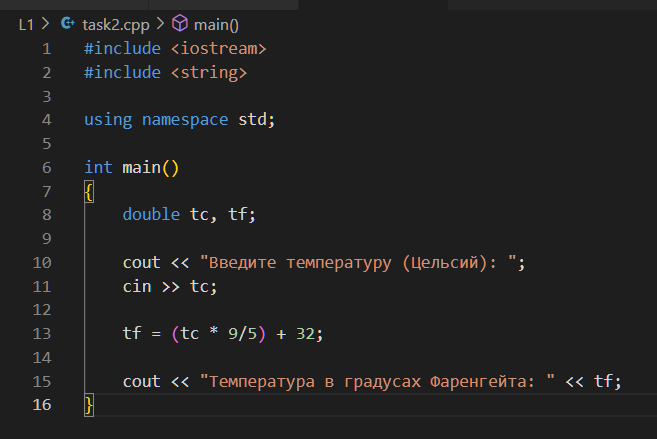


Рисунок - L1 Task2

Результат работы программы:

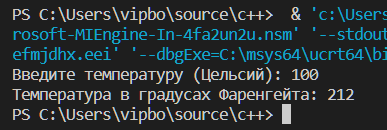


Рисунок результат L1 Task2

Задание №3: напишите программу, запрашивающую у пользователя значения двух дробей, а затем выводящую результат, также записанный в форме дроби.

Решение:

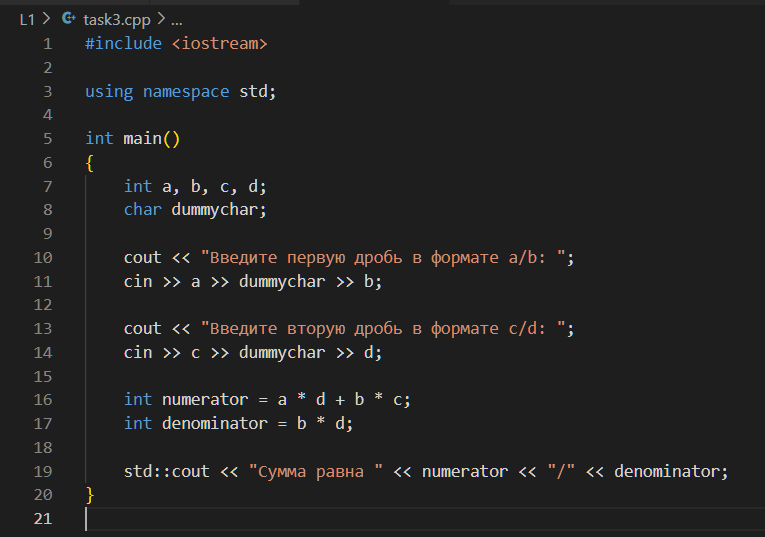


Рисунок - L1 Task3

Результат работы программы:

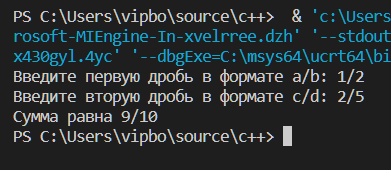


Рисунок – результат L1 Task3

Задание №4: напишите программу, которая будет преобразовывать сумму, записанную в старом формате (фунты, шиллинги, пенсы), в новый формат (фунты, пенсы).

Решение:

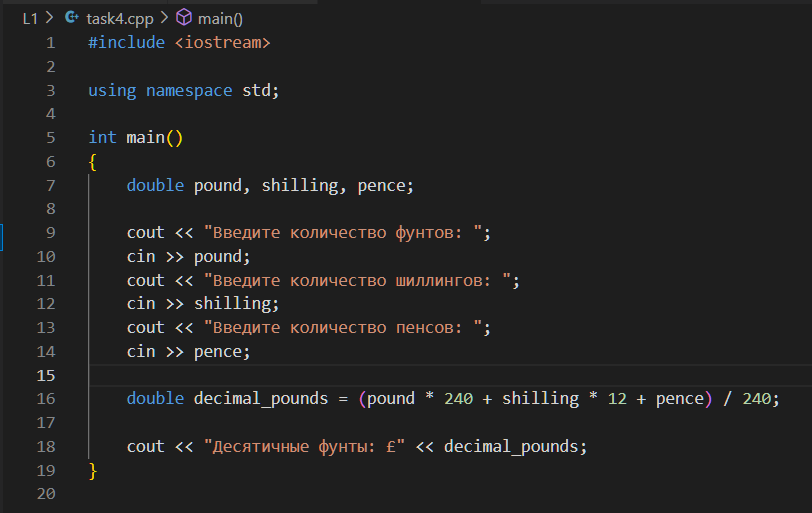


Рисунок – L1 Task4

Результат работы программы:

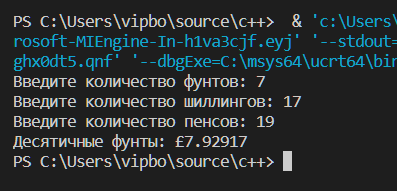


Рисунок – результат L1 Task4

Задание №5: напишите программу, выполняющую действия, обратные тем, которые описаны в предыдущем упражнении 4, то есть запрашивающую у пользователя сумму, указанную в десятичных фунтах, и переводящую ее в старую систему.

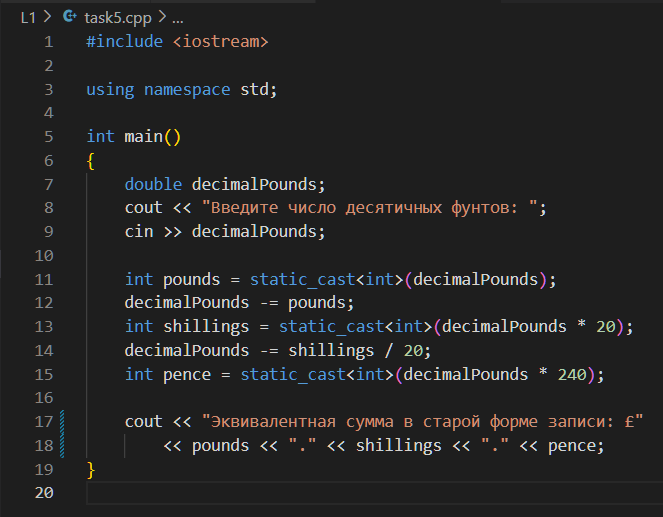


Рисунок – L1 Task5

Результат работы программы:

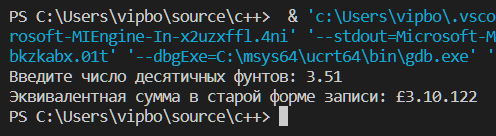


Рисунок – результат L1 Task5

Вывод: в ходе практической работы были получены практические навыки в использовании языка c++ за счет решения задач.

Практическая работа №2 «Циклы и ветвления»

Цель работы: получить практические навыки в создании программ с циклами и ветвлениями.

Задание №1: при помощи цикла for изобразите на экране пирамиду из символов 'X'.

Решение:

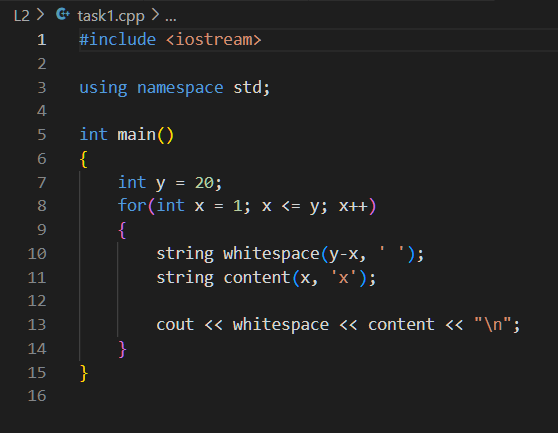


Рисунок – L2 Task1

Результат работы программы:

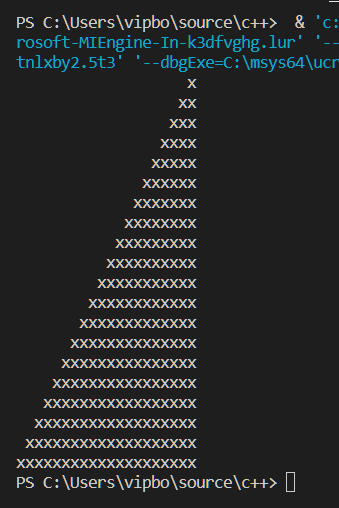


Рисунок – результат L2 Task1

Задание №2: напишите программу вычисления факториала таким образом, чтобы она циклически запрашивала ввод пользователем числа и вычисляла его факториал, пока пользователь не введет 0. В этом случае программа должна завершиться. Вычисление должно строиться на основе использования цикла.

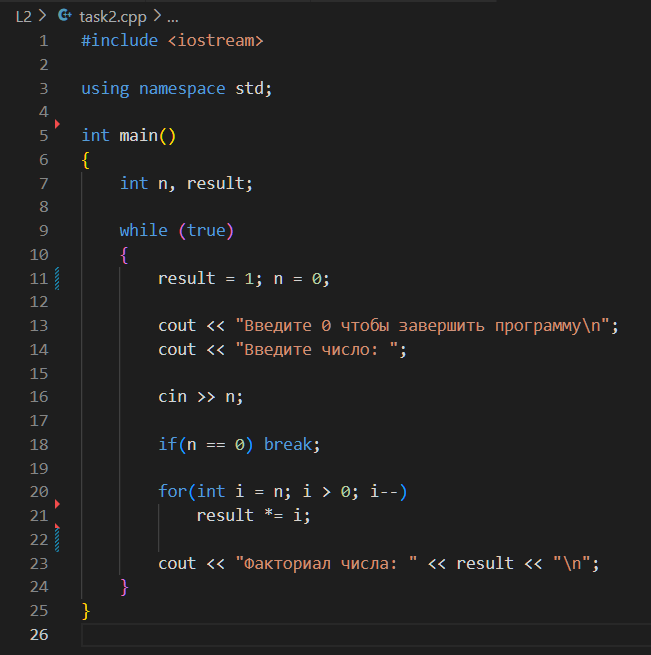


Рисунок – L2 Task2

Результат работы программы:

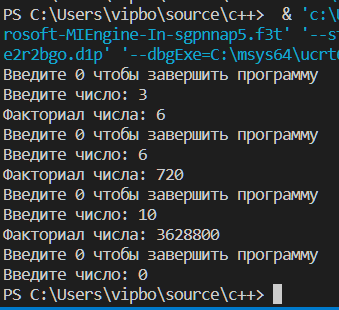


Рисунок - результат L2 Task2

Задание №3: Напишите программу, рассчитывающую сумму денег, которые вы получите при вложении начальной суммы с фиксированной процентной ставкой дохода через определенное количество лет. Пользователь должен вводить с клавиатуры начальный вклад, число лет и процентную ставку.

Решение:

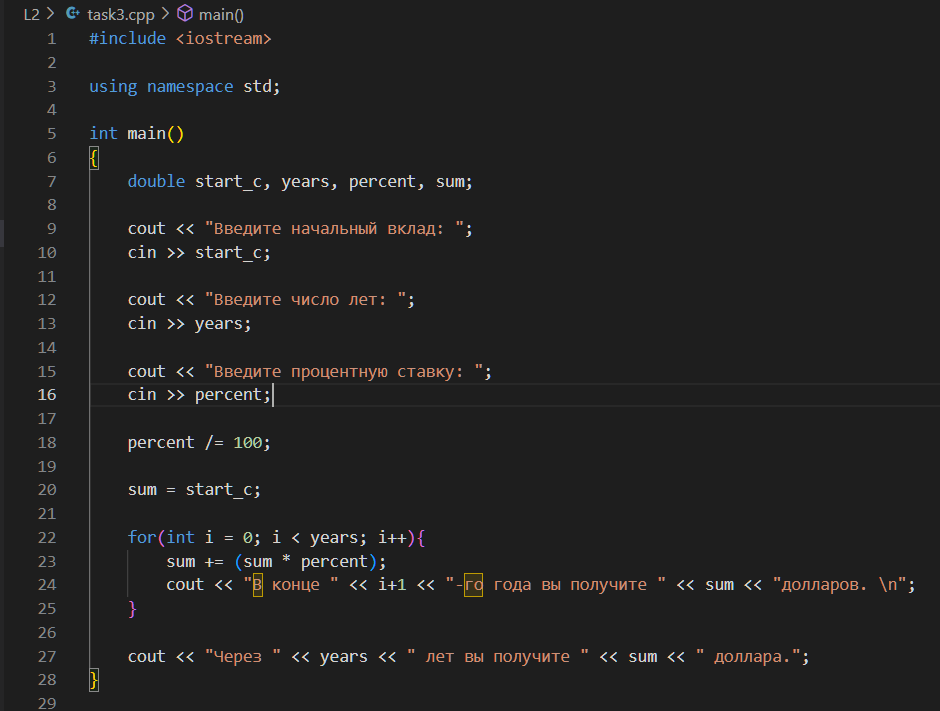


Рисунок – L2 Task3

Результат работы программы:

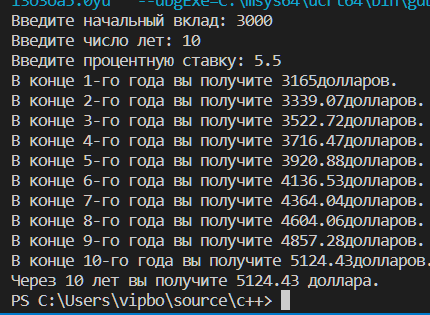


Рисунок – результат L2 Task3

Задание №4: напишите программу, рассчитывающую сумму денег, которые придется выплатить при покупке квартиры в ипотеку с фиксированной процентной ставкой к моменту окончания всех платежей. Пользователь должен вводить с клавиатуры стоимость квартиры, начальный взнос, число лет ипотеки и процентную ставку. На экран вывести объемы платежей по годам и общую сумму выплат.

Решение:

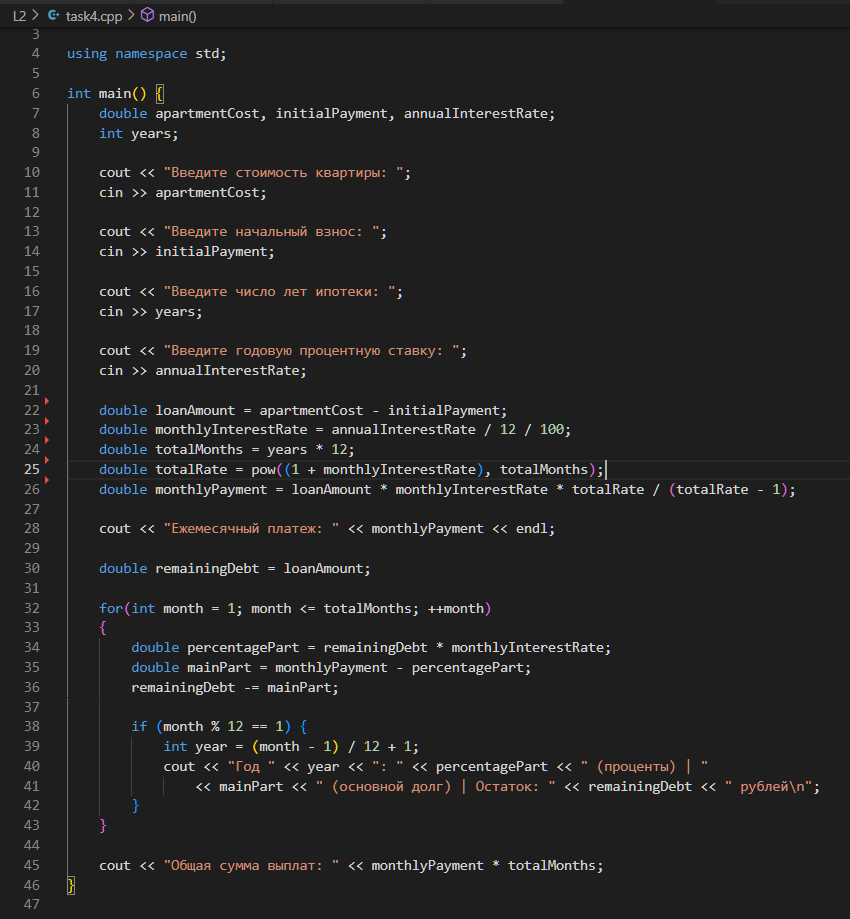


Рисунок 17 L2 Task4

Результат работы программы:

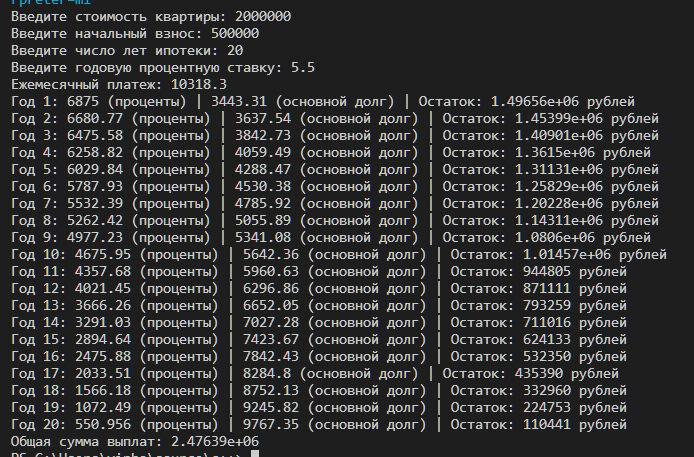


Рисунок - результат L2 Task4

Задание №5: напишите программу, которая циклически будет запрашивать ввод пользователем двух денежных сумм, выраженных в фунтах, шиллингах и пенсах (см. упражнения предыдущей ЛР). Программа должна складывать введенные суммы и выводить на экран результат, также выраженный в фунтах, шиллингах и пенсах. После каждой итерации программа должна спрашивать пользователя, желает ли он продолжать работу программы

Решение:

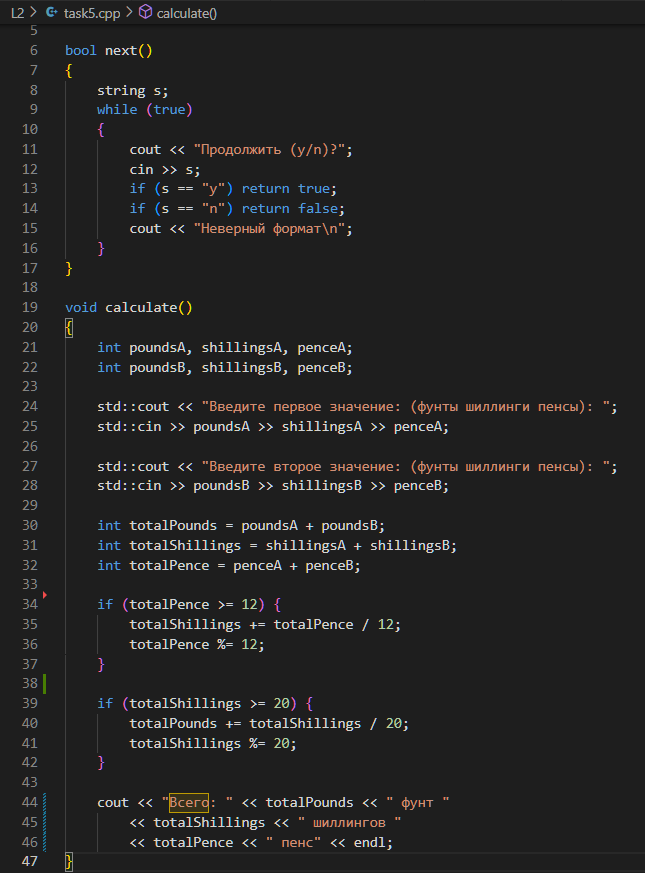


Рисунок 19 – L2 Task5 Методы next и calculate

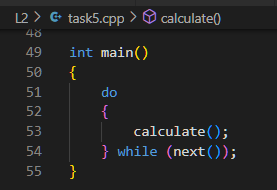


Рисунок - L2 Task5

Результат работы программы:

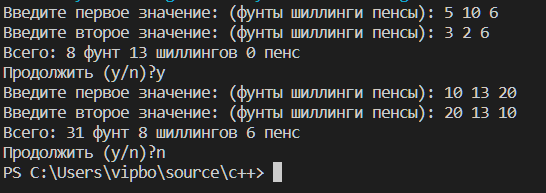


Рисунок – результат L2 Task5

Вывод: в ходе практической работы были получены практические навыки в создании программ с циклами и ветвлениями.

Практическая работа №3 «Структуры и перечисления»

Цель работы: получить практические навыки в создании программ, использующих структуры и перечисления.

Задание №1: создайте структуру с именем employee, содержащую два поля: номер сотрудника типа int и величину его зарплаты в рублях типа float. Запросите с клавиатуры данные о трех сотрудниках, сохраните их в трех структурных переменных типа employee и выведите информацию о каждом из сотрудников на экран.

Решение:

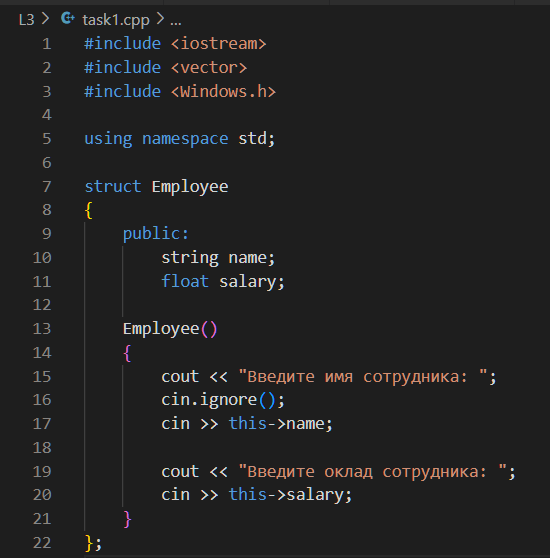


Рисунок – L3 Task1 (1)

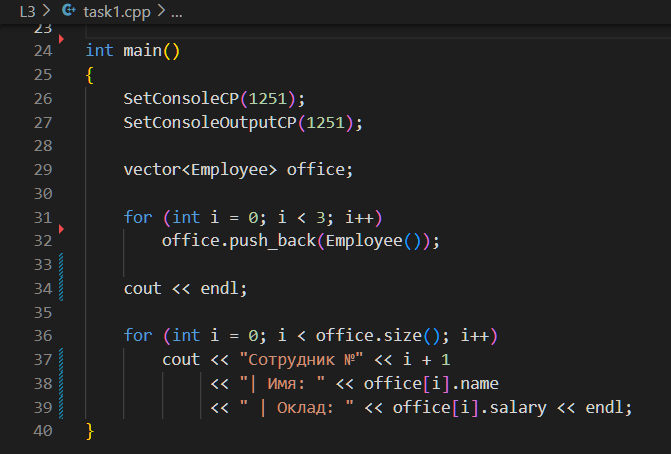


Рисунок - L3 Task1 (2)

Результат выполнения программы:

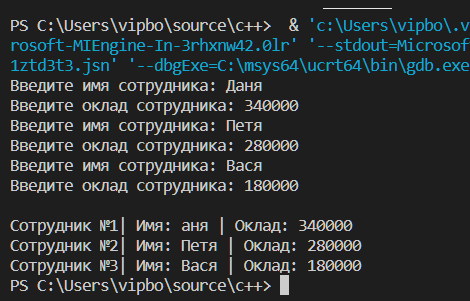


Рисунок – результат L3 Task1

Задание №2: Пусть, например, в программе определен перечисляемый тип данных etype, отражающий должность сотрудника: enum etype { laborer, secretary, manager, accountant, executive, researcher };

Напишите программу, которая сначала по первой букве должности, введенной пользователем, определяет соответствующее значение переменной, помещает это значение в переменную типа etype, а затем выводит полностью название должности, первую букву которой ввел пользователь.

Решение: в данном решении я не использовал switch (надеюсь меня простят)

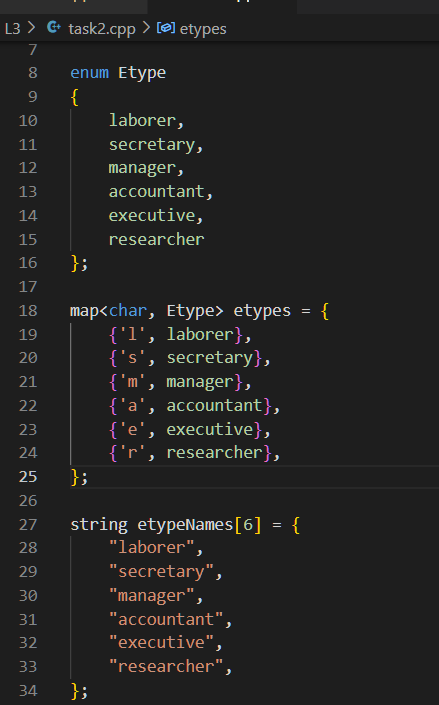


Рисунок – L3 Task2 данные

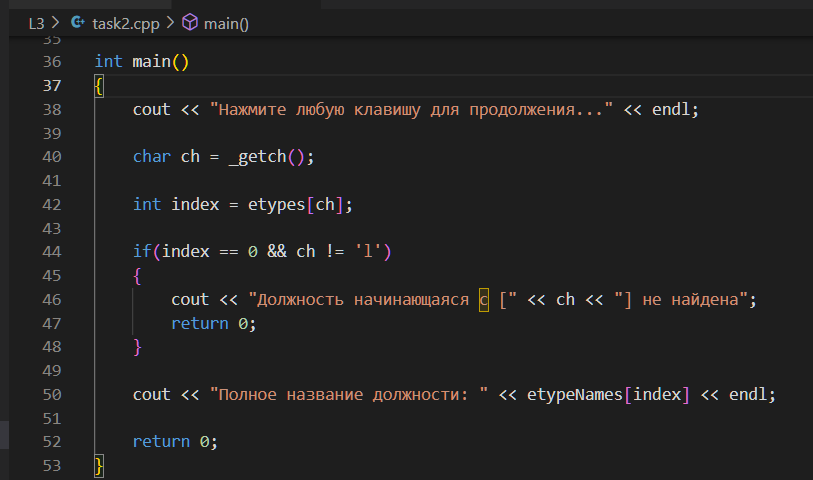


Рисунок – L3 Task2 main

Результат работы программы:



Рисунок – L3 Task2

Задание №3: Вернитесь к упражнению 3 ЛР1. В этом упражнении требуется написать программу, которая хранит значения двух дробей в виде числителя и знаменателя, а затем складывает эти дроби согласно арифметическому правилу. Измените эту программу так, чтобы значения дробей хранились в структуре fraction, состоящей из двух полей типа int. предназначенных для хранения числителя и знаменателя. Все значения дробей должны храниться в переменных типа fraction.

Решение:

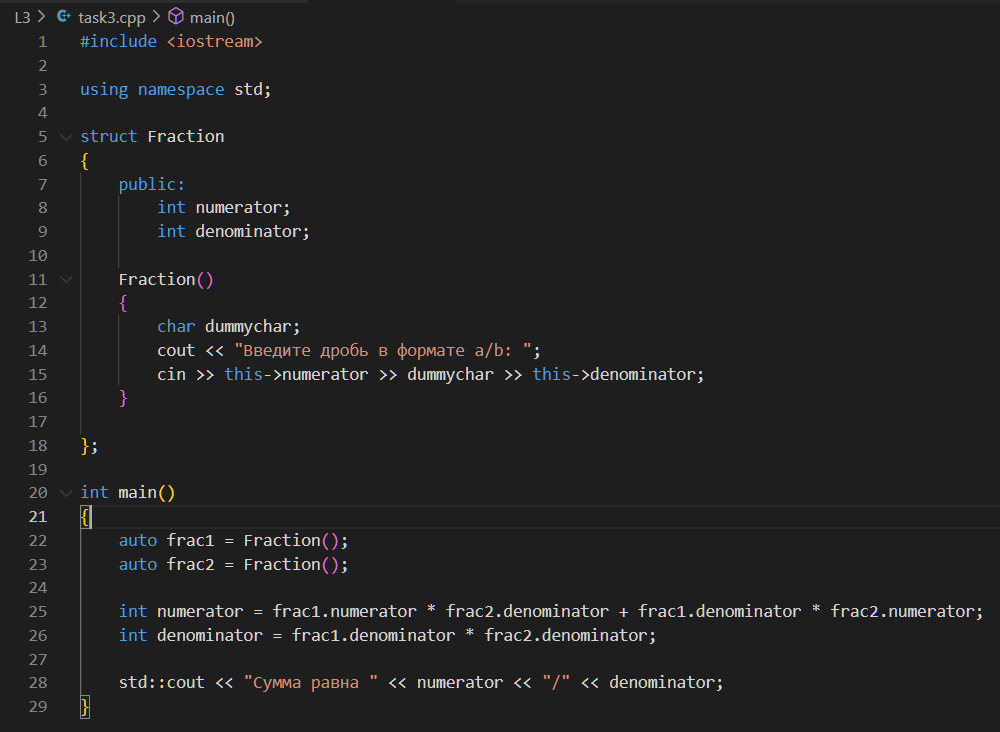


Рисунок – L3 Task3

Результат работы программы:

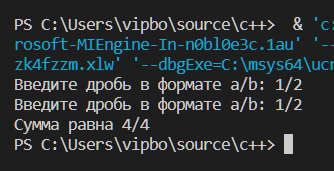


Рисунок – результат L3 Task3

Задание №4: Создайте структуру с именем time. Три ее поля, имеющие тип int. будут называться hours, minutes и seconds. Напишите программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Программа должна хранить время в структурной переменной типа time и выводить количество секунд в введенном времени.

Решение:

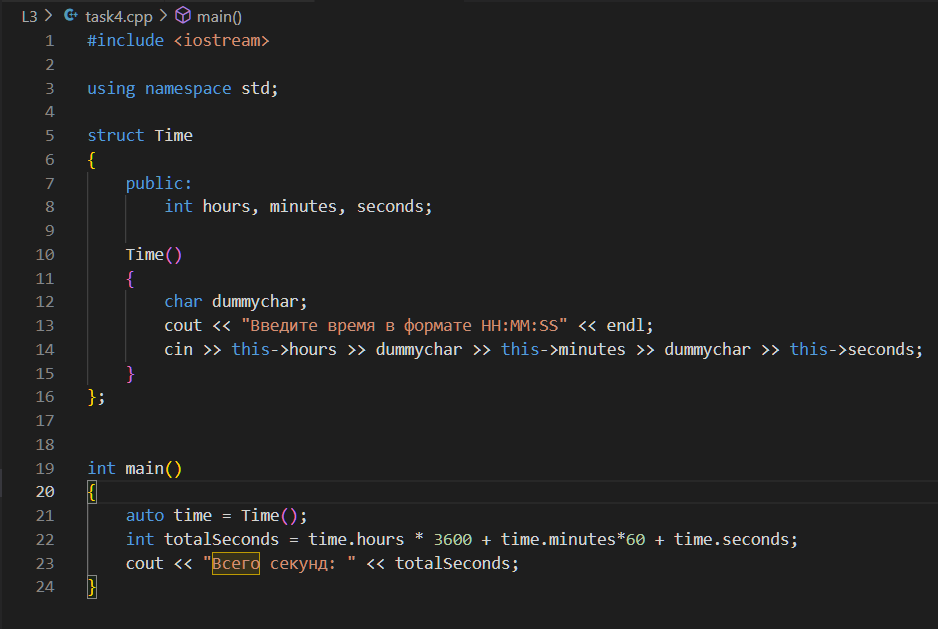


Рисунок – L3 Task4

Результат работы программы:

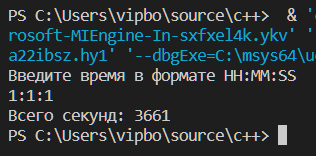


Рисунок – результат L3 Task4

Задание №5: Используя структуру time из предыдущего упражнения, напишите программу, которая получает от пользователя два значения времени в формате 12:59:59, сохраняет их в переменных типа struct time, затем переводит оба значения в секунды, складывает их, переводит сумму в исходный формат, сохраняет его в переменной типа time и выводит полученный результат на экран в формате 12:59:59.

Решение:

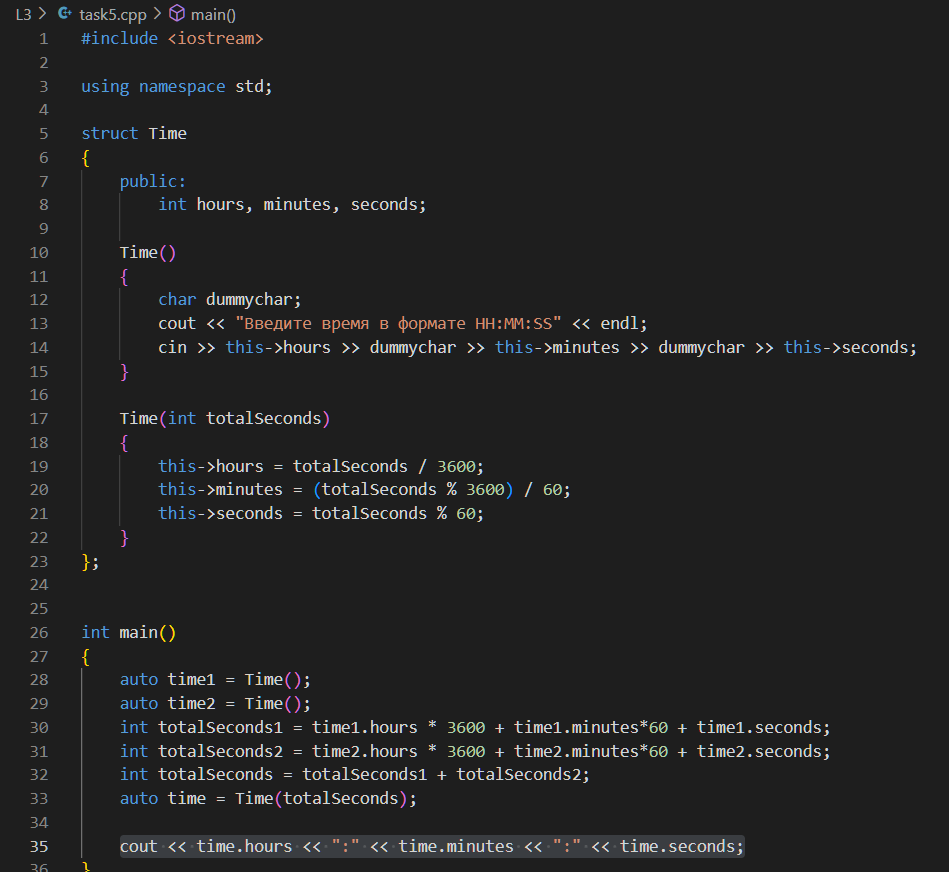


Рисунок – L3 Task5

Результат работы программы:

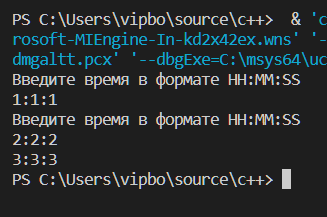


Рисунок – результат L3 Task5

Вывод: в ходе практической работы были получены практические навыки в создании программ, использующих структуры и перечисления.

Практическая работа №4 «Функции»

Цель работы: получить практические навыки в создании функций и их последующем использовании.

Задание №1: Напишите функцию с именем hms\_to\_secs(), имеющую три аргумента типа int: часы, минуты и секунды. Функция должна возвращать эквивалент переданного ей временного значения в секундах (типа long). Создайте программу, которая будет циклически запрашивать у пользователя ввод значения часов, минут и секунд и выводить результат работы функции на экран.

Решение: функция next() была ранее использована в L2 Task5

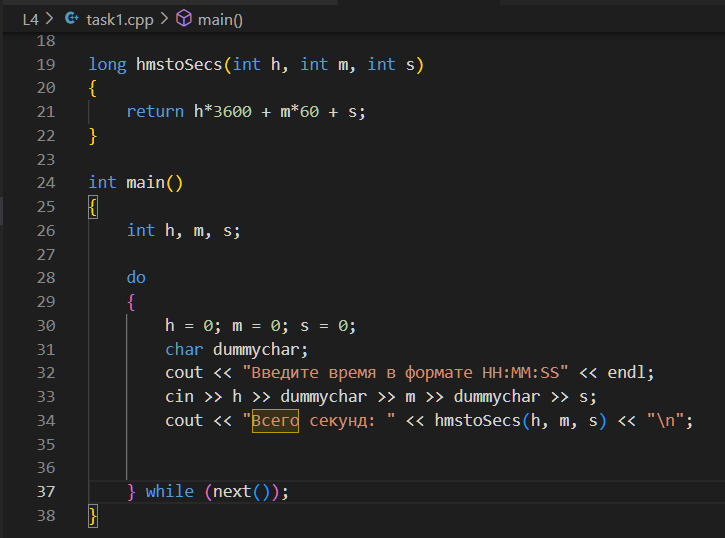


Рисунок – L4 Task1

Результат работы программы:

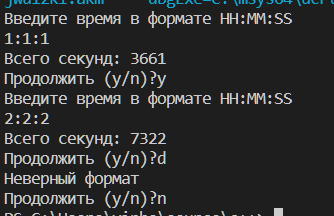


Рисунок – результат L4 Task1

Задание №2: модифицируйте программу, описанную в упражнении 5 предыдущей лабораторной работы «Структуры и перечисления», складывающую два структурных значения типа time. Теперь программа должна включать в себя две функции. Первая, time\_to\_secs(), принимает в качестве аргумента значение типа time и возвращает эквивалентное значение в секундах типа long. Вторая, secs\_to\_time(), в качестве аргумента принимает число секунд, имеющее тип long, а возвращает эквивалентное значение типа time.

Решение:

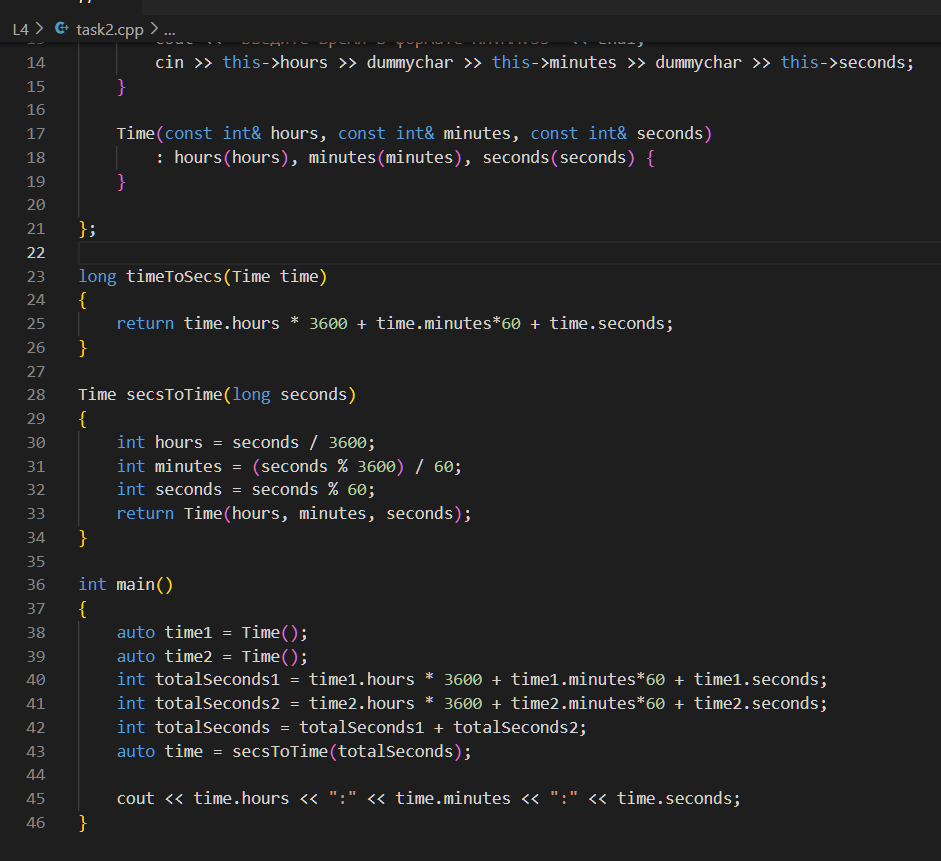


Рисунок – L4 Task2

В данном решении используется структура из L3 Task4 с дополнительным конструктором, который принимает аргументы hours, minutes и seconds.

Результат работы программы:

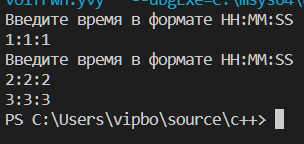


Рисунок – результат L4 Task2

Задание №3: напишите функцию с именем swap(), обменивающую значениями два своих аргумента типа int (обратите внимание, что изменяться должны значения переменных из вызывающей функции, а не локальных переменных функции). Выберите способ передачи аргументов. Напишите вызывающую программу main(), использующую данную функцию.

Решение:

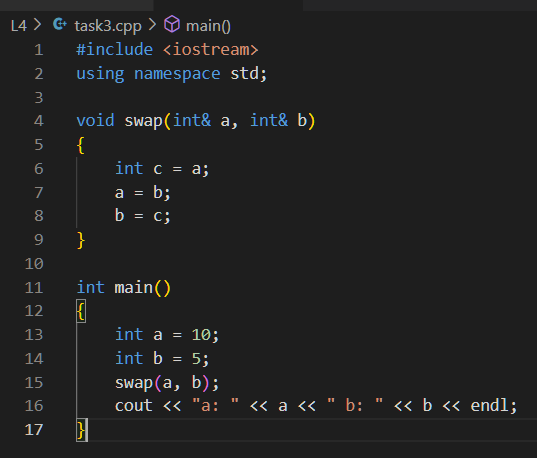


Рисунок – L3 Task3

Результат работы программы:

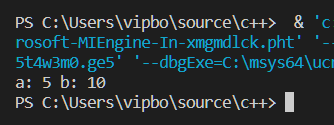


Рисунок – результат L4 Task3

Задание №4: переработайте программу из предыдущего упражнения 3 так, чтобы функция swap() принимала в качестве аргументов значения типа time (см. упражнение 2).

Решение: данное решение было реализовано с помощью дженериков. А в структуру Time был добавлен метод toString().

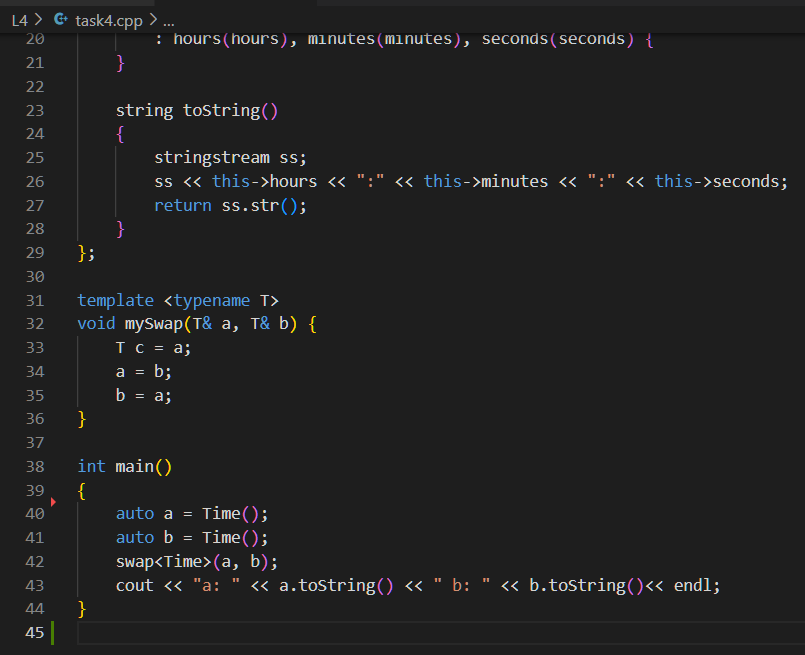


Рисунок – L4 Task4

Результат работы программы:

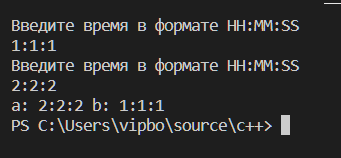


Рисунок – результат L4 Task4

Задание №5: напишите функцию, которая при каждом вызове будет выводить на экран количество раз, которое она вызывалась ранее. Напишите программу, которая будет вызывать данную функцию не менее 10 раз. Попробуйте реализовать данную функцию двумя различными способами: с использованием глобальной переменной и статической локальной переменной для хранения числа вызовов функции.

Решение:

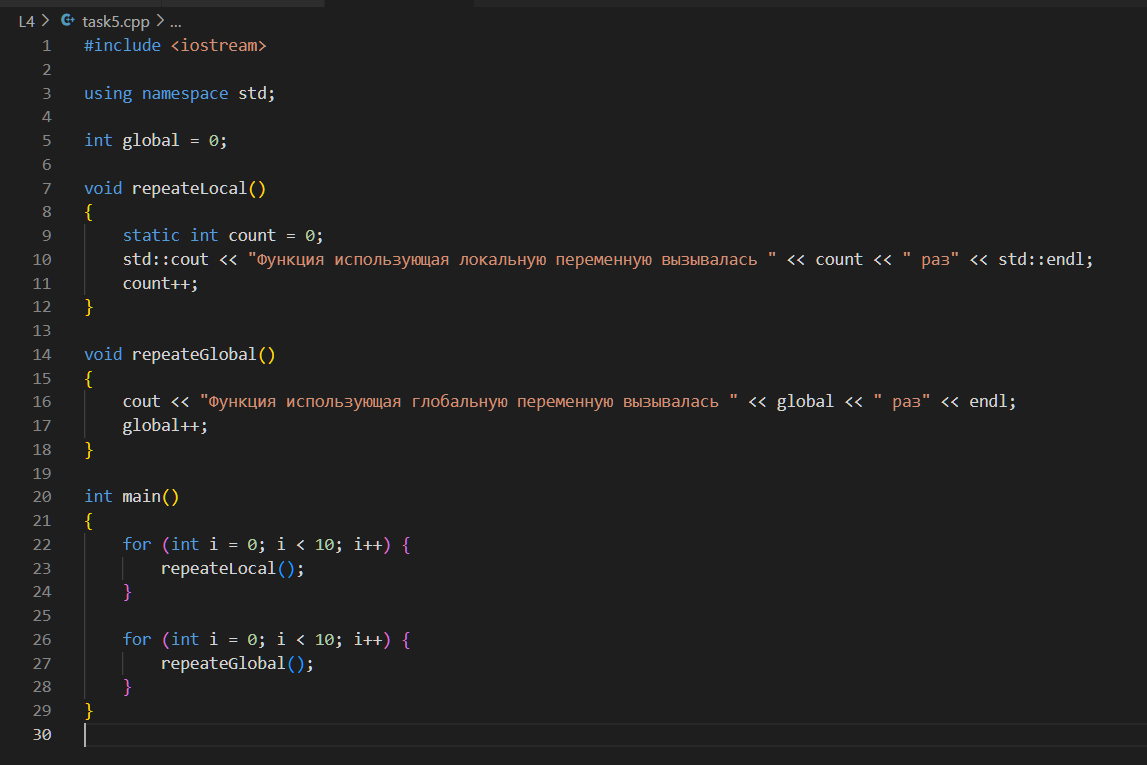


Рисунок – L4 Task5

Результат работы программы:

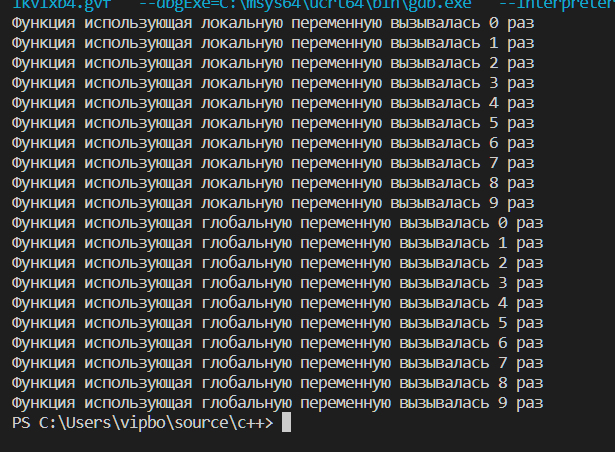


Рисунок – результат L4 Task5

Вывод: в ходе практической работы получены практические навыки в создании функций и их последующем использовании.