| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
| --- | --- | --- |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №2  по дисциплине «Языки программирования»  Вариант 8 | | |
|  | | Работу выполнил  студент группы ФИТ-3,4-2023 2 курса  Рычагов М.Р.  «11» Октября 2024 г. |
| Работу проверила  Кнутова Н.С  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

**СОДЕРЖАНИЕ**

[Задание 1 4](#_heading=h.30j0zll)

[Текст задания 4](#_heading=h.1fob9te)

[Алгоритм решения 4](#_heading=h.3znysh7)

[Тестирование 4](#_heading=h.2et92p0)

[Код программы 4](#_heading=h.tyjcwt)

[Задание 2 5](#_heading=h.3dy6vkm)

[Текст задания 5](#_heading=h.1t3h5sf)

[Алгоритм решения 5](#_heading=h.4d34og8)

[Тестирование 5](#_heading=h.2s8eyo1)

[Код программы 5](#_heading=h.17dp8vu)

[Задание 3 6](#_heading=h.3rdcrjn)

[Текст задания 6](#_heading=h.26in1rg)

[Алгоритм решения 6](#_heading=h.lnxbz9)

[Тестирование 6](#_heading=h.35nkun2)

[Код программы 6](#_heading=h.44sinio)

[Задание 4 7](#_heading=h.2jxsxqh)

[Текст задания 7](#_heading=h.z337ya)

[Алгоритм решения 7](#_heading=h.3j2qqm3)

[Тестирование 7](#_heading=h.1y810tw)

[Код программы 7](#_heading=h.4i7ojhp)

[Задание 5 8](#_heading=h.2xcytpi)

[Текст задания 8](#_heading=h.1ci93xb)

[Алгоритм решения 8](#_heading=h.3whwml4)

[Тестирование 8](#_heading=h.2bn6wsx)

[Код программы 8](#_heading=h.qsh70q)

[Задание 6 9](#_heading=h.3as4poj)

[Текст задания 9](#_heading=h.1pxezwc)

[Алгоритм решения 9](#_heading=h.49x2ik5)

[Тестирование 9](#_heading=h.2p2csry)

[Код программы 9](#_heading=h.3o7alnk)

[Задание 7 10](#_heading=h.23ckvvd)

[Текст задания 10](#_heading=h.ihv636)

[Алгоритм решения 10](#_heading=h.32hioqz)

[Тестирование 10](#_heading=h.1hmsyys)

[Код программы 10](#_heading=h.41mghml)

[Задание 8 11](#_heading=h.2grqrue)

[Текст задания 11](#_heading=h.vx1227)

[Алгоритм решения 11](#_heading=h.3fwokq0)

[Тестирование 11](#_heading=h.1v1yuxt)

[Код программы 11](#_heading=h.2u6wntf)

[Задание 9 12](#_heading=h.19c6y18)

[Текст задания 12](#_heading=h.3tbugp1)

[Алгоритм решения 12](#_heading=h.28h4qwu)

[Тестирование 12](#_heading=h.nmf14n)

[Код программы 12](#_heading=h.37m2jsg)

[Задание 10 13](#_heading=h.1mrcu09)

[Текст задания 13](#_heading=h.46r0co2)

[Алгоритм решения 13](#_heading=h.2lwamvv)

[Тестирование 13](#_heading=h.111kx3o)

[Код программы 13](#_heading=h.3l18frh)

# Задание 1

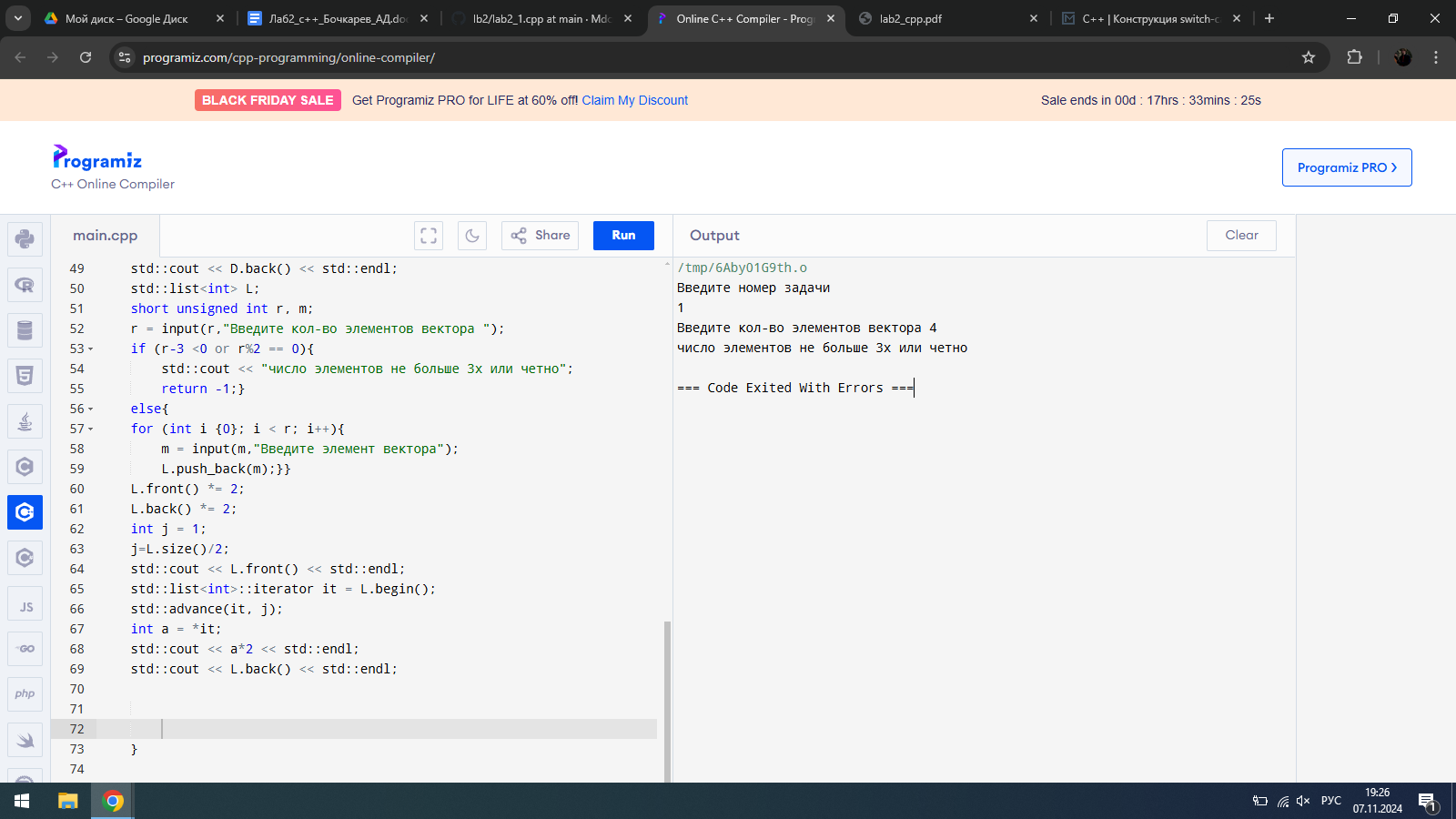
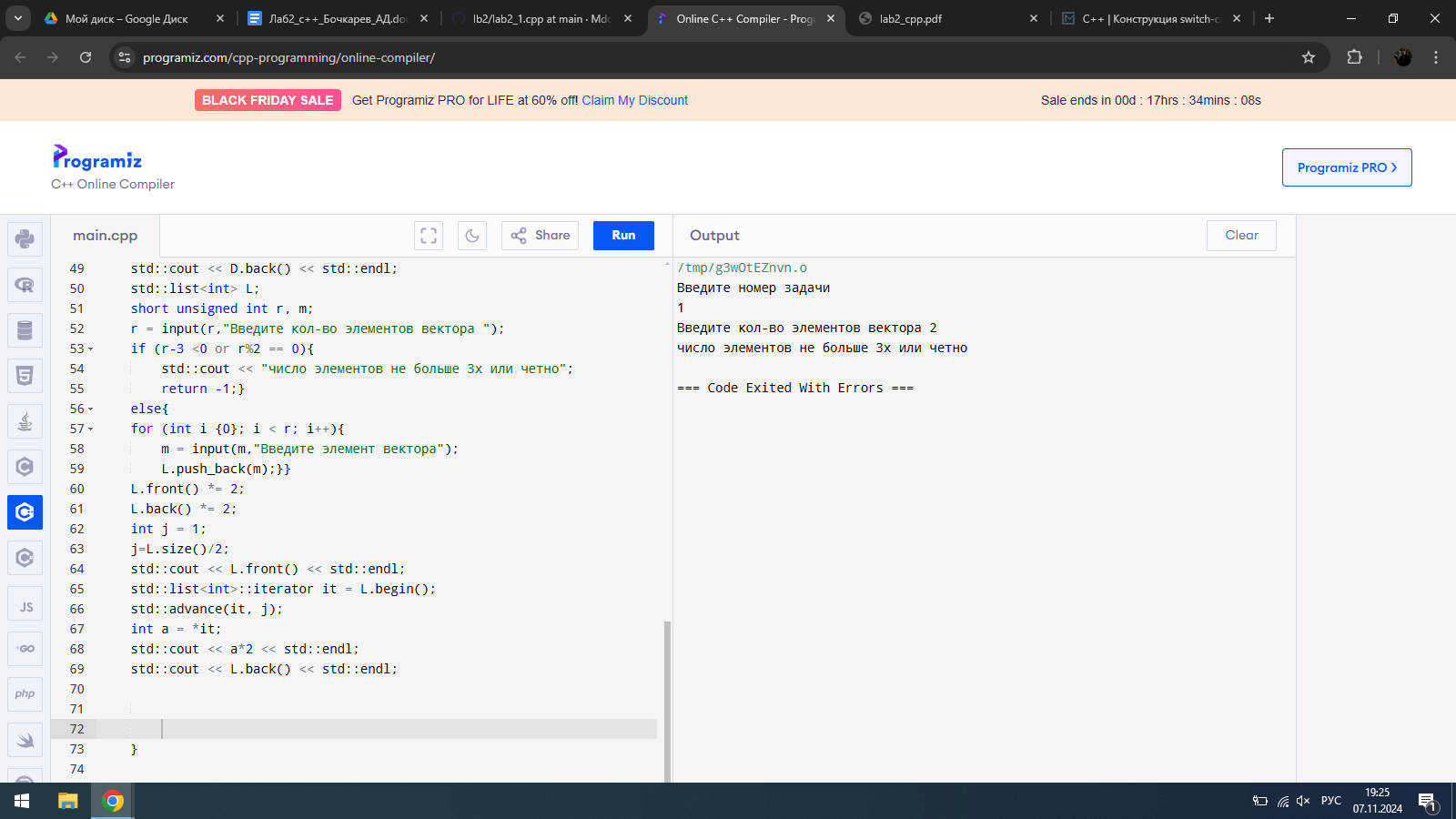
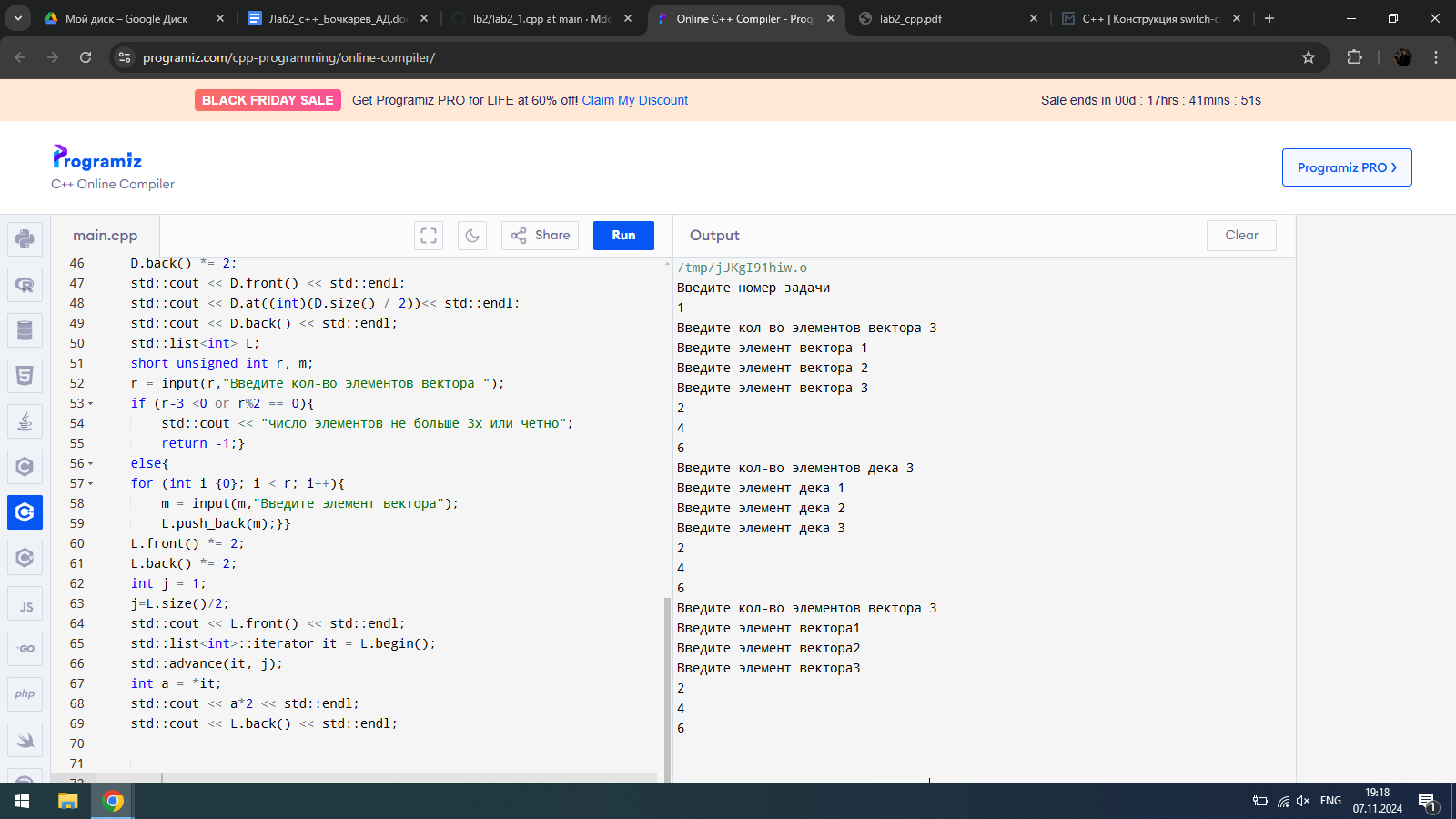
## Текст задания

Даны вектор V, дек D и список L. Каждый исходный контейнер содержит не менее трех элементов, количество элементов является нечетным. Удвоить значения первого, среднего и последнего элемента каждого из исходных контейнеров.

## Алгоритм решения

Вводим с клавиатуры значения вектора, дека и списка.В случаях дека и вектора просто обращаюсь по индексу к первому, V.size\D.size /2, к последнему элементу и домножаю каждое значение на 2. В случае списка сдвигаю значение итератоа до искомой середины списка и домножаю на 2. первый и последний элемент также умножаются на 2.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_1.cpp>

# Задание 2

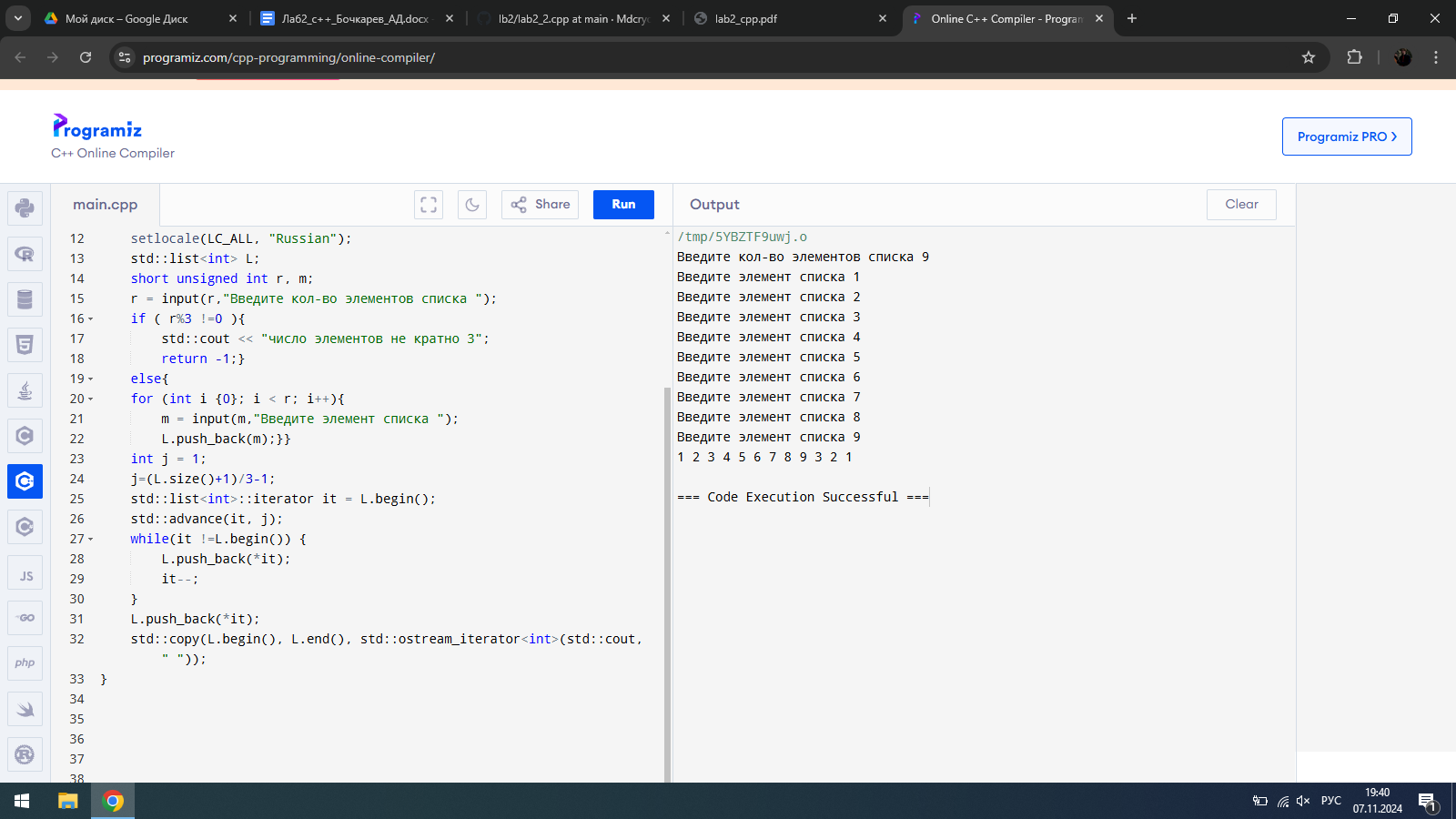
## Текст задания

Дан список L, количество элементов которого делится на 3. Добавить в конец списка первую треть его исходных элементов в обратном порядке. Использовать один вызов функции-члена insert.

## Алгоритм решения

Вводим количество элементов в списке L. Сдвигаем значение итератора до границы, т.е. элемента с индексом (L.size()+1)/3. после чего итератор проходить цикл до L.begin(), добавляя значения в конец массива. последнее значение добавляется вне цикла.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_2.cpp>

# Задание 3

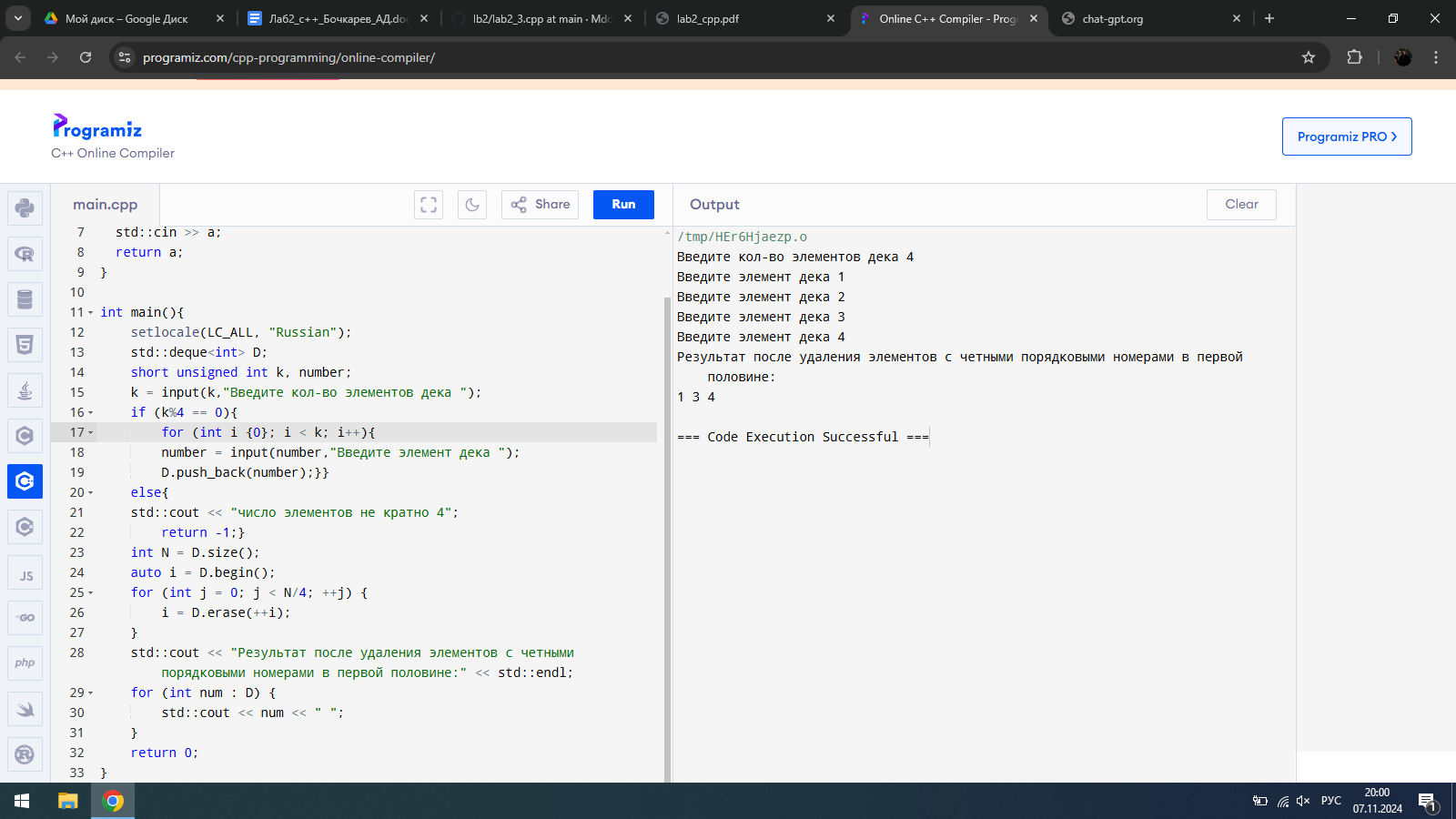
## Текст задания

Дан дек D с количеством элементов, кратным 4. Удалить в первой половине исходного дека все элементы с четными порядковыми номерами (считая, что начальный элемент дека имеет порядковый номер 1). Использовать функцию-член erase в цикле с числовым параметром. Указание. Используйте цикл с числовым параметром, повторяющийся N/4 раз (где N — исходный размер дека). Свяжите вспомогательный итератор i с началом дека. В цикле выполняйте функцию-член erase с параметром, равным ++i, обязательно присваивая возвращаемое значение итератору i.

## Алгоритм решения

Вводим количество элементов в деке, даем вспомогательному итератору i значение D.begin(). Проходим деку ¼ раза. Каждый раз когда значение счетчика j увеличивается мы передаем i удаление следующего после него значения, таким образом удаляя из первой половины деки четные по индексу элементы.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_3.cpp>

# Задание 4

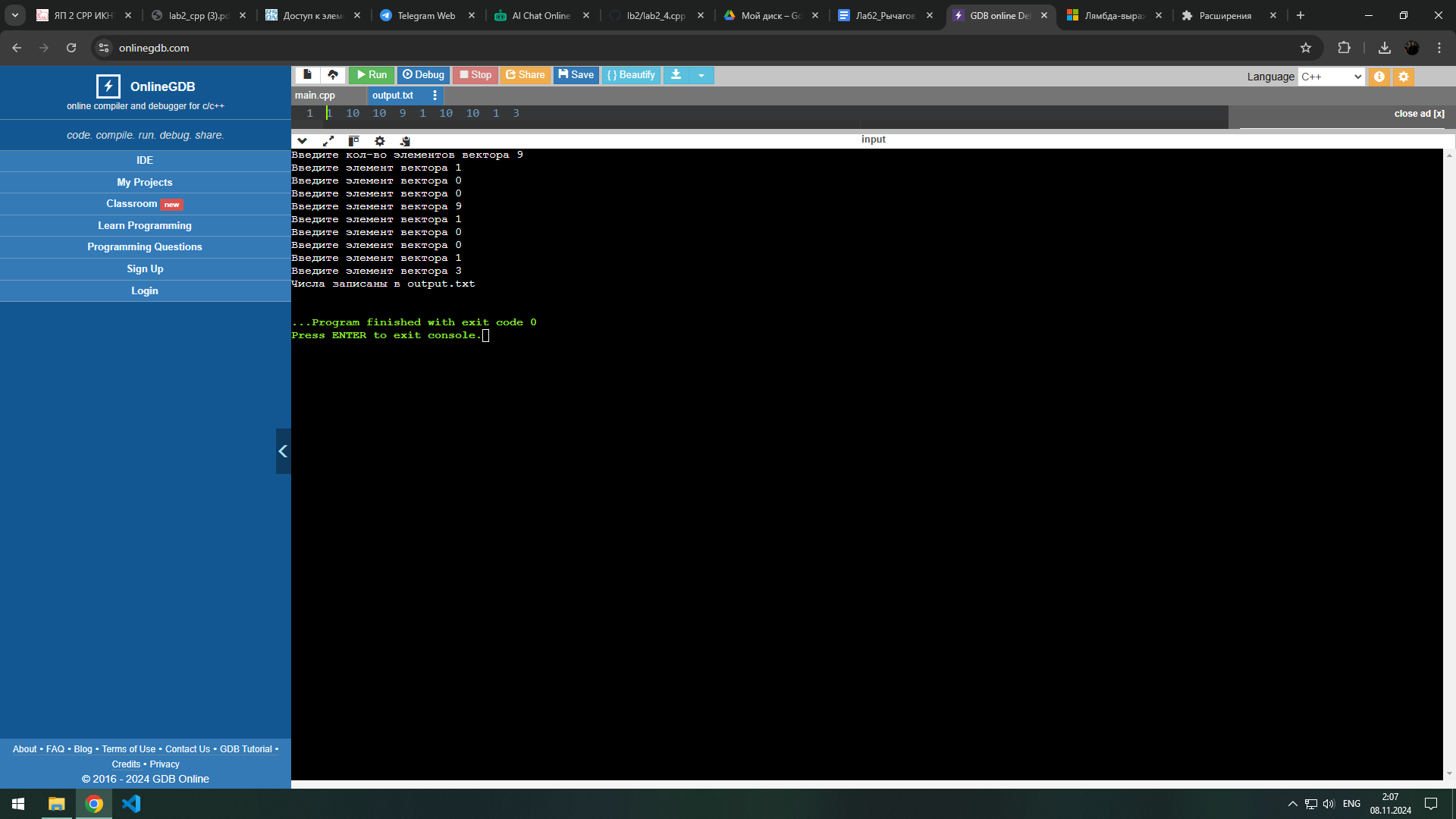
## Текст задания

Дана строка name и набор целых чисел. Записать в текстовый файл с именем name все числа из исходного набора в том же порядке, заменяя каждое число 0 на число 10 и добавляя после каждого числа два пробела. Использовать алгоритм replace\_copy.

## Алгоритм решения

Создаем файл, вводим элементы, в алгоритме Replace\_copy, прописываем интервал на котором будем заменять и значения, которые будем менять и на которые будем менять.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_4.cpp>

# Задание 5

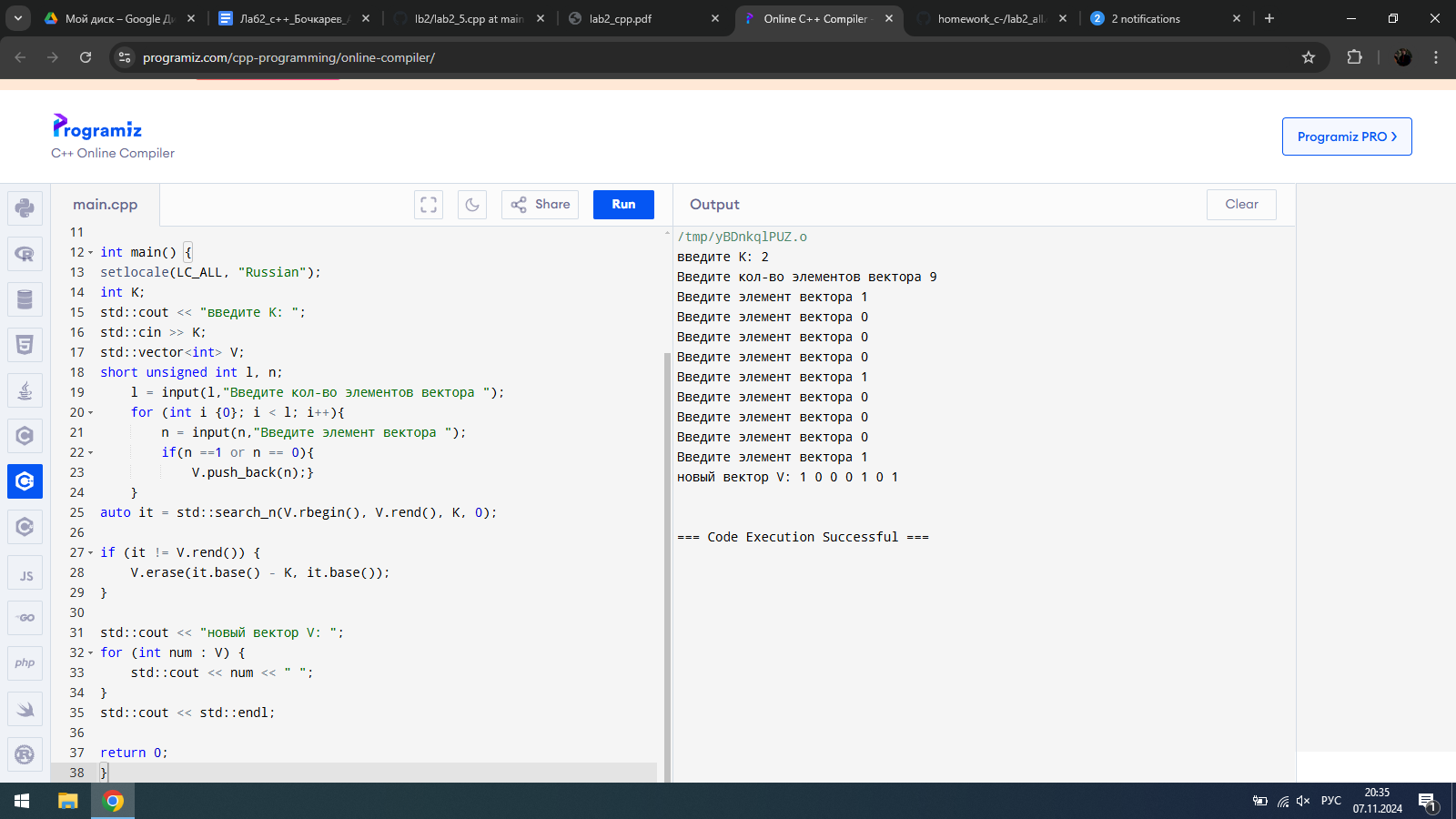
## Текст задания

Дано целое число K (> 0) и вектор V, содержащий только нули и единицы. Удалить в векторе V последний набор из K подряд расположенных нулей (если в этом наборе имеется больше K нулей, то требуется удалить только последние K из них). Если вектор не содержит требуемого набора нулей, то не изменять его. Использовать алгоритм search\_n и функцию-член erase, а также обратные итераторы.

## Алгоритм решения

Вводим элементы в вектор и число K. Создаем итератор, который проходит через вектор пока не найдет последние более чем K нулей. После чего итератор удаляет последние K идущих подряд нулей.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_5.cpp>

# Задание 6

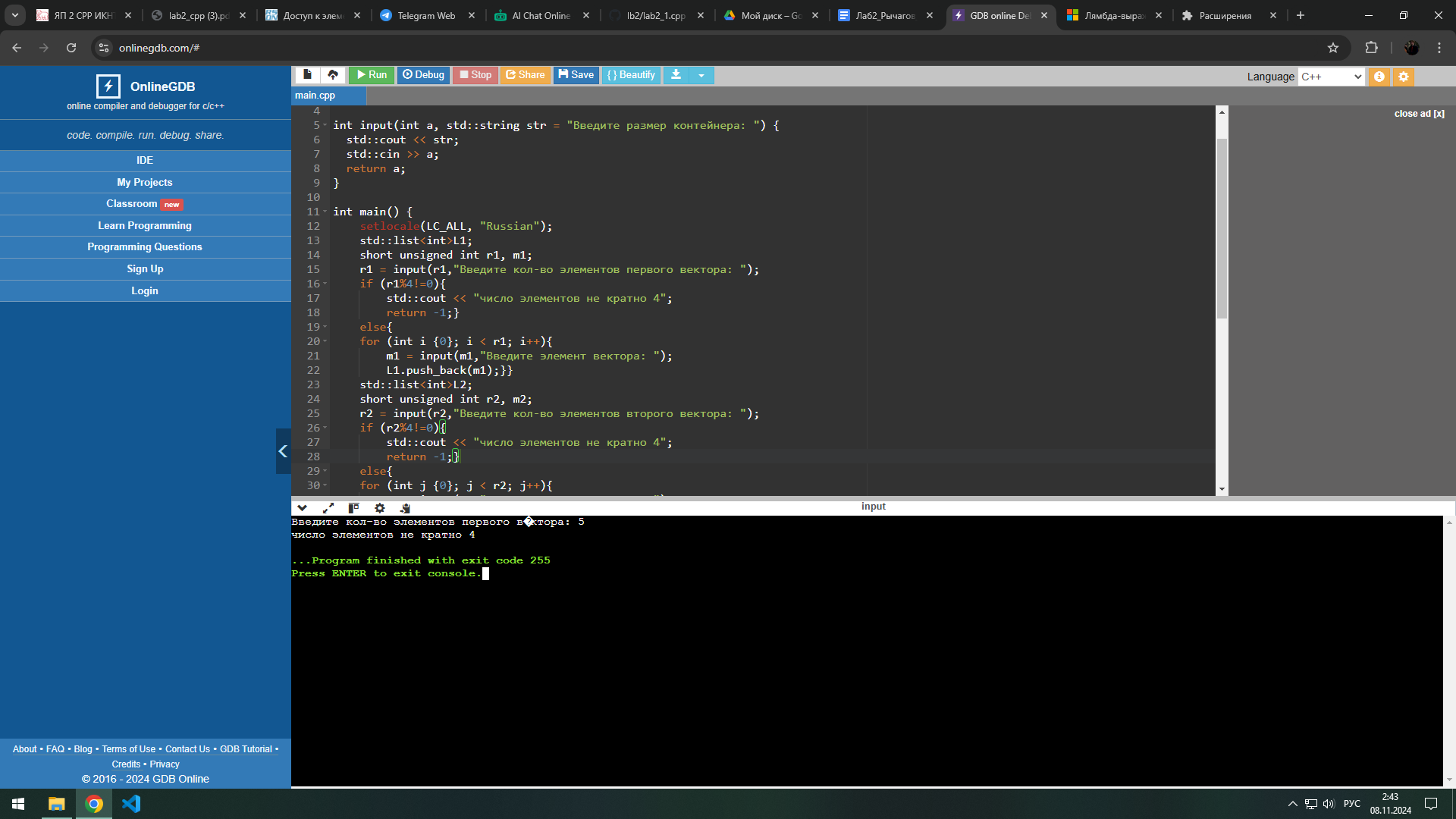
## Текст задания

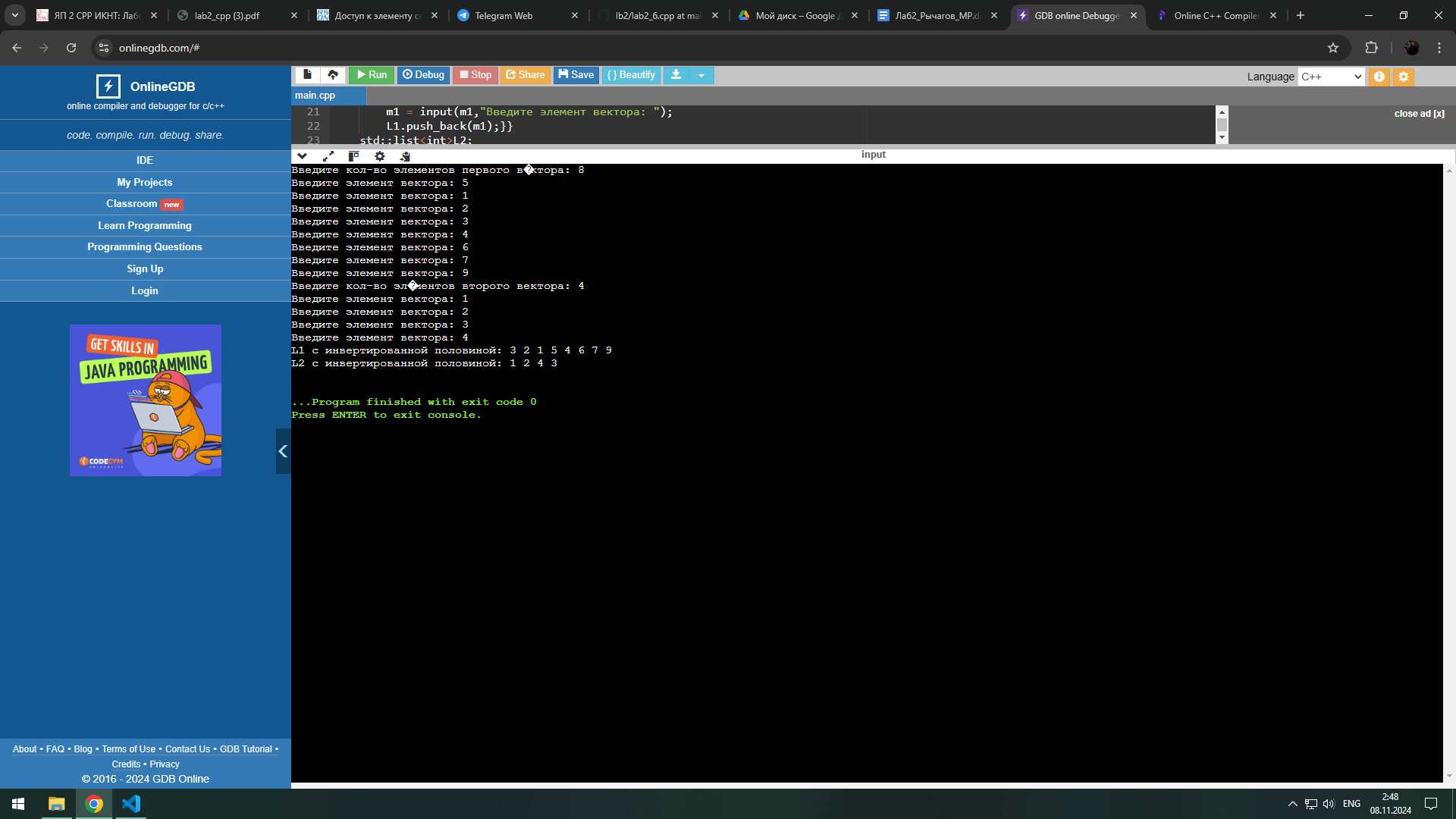
Даны списки L1 и L2, у каждого из которых количество элементов делится на 4. В первом списке инвертировать (расположить в обратном порядке) первую половину элементов, во втором списке — вторую половину. Для первого списка использовать алгоритм swap\_ranges и обратные итераторы, для второго — алгоритм reverse. Использовать также функцию advance.

## Алгоритм решения

Вводим с клавиатуры значения для списков L1 и L2. Определяем границу по середине списка. Проходим алгоритмом реверс от начала(для первого) и с конца(для второго) до границы. Выводим результат

## Тестирование





## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_10.cpp>

# Задание 7

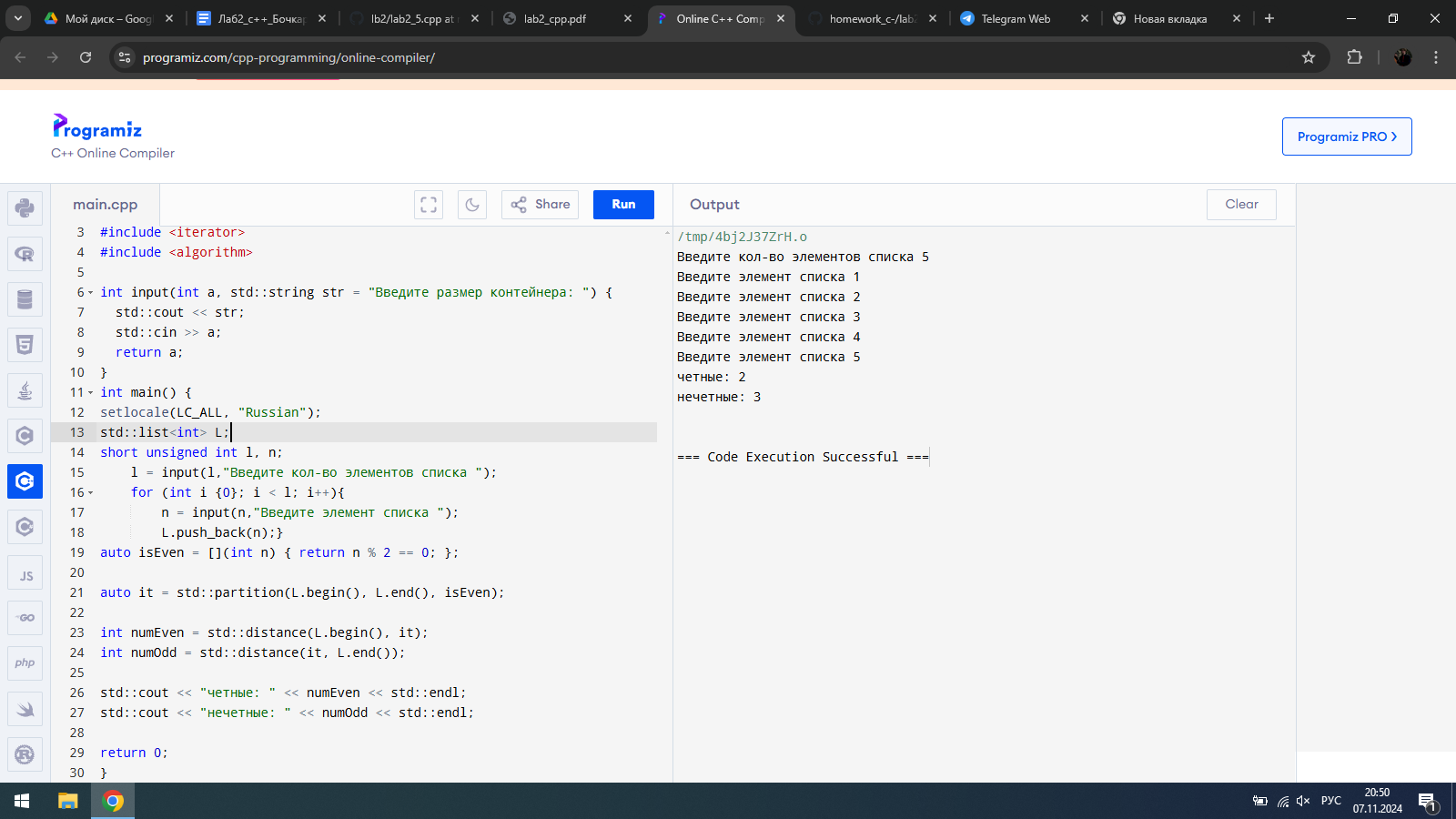
## Текст задания

Дан список L. Определить количество четных и нечетных чисел в исходном списке (вначале вывести количество четных, затем количество нечетных чисел). Использовать алгоритм partition и два вызова функции distance для итераторов.

## Алгоритм решения

Вводим значения списка, после чего при помощи лямбда функции перемещаем четные значения в начало списка, а нечетные в конец. Для подсчетов используем итераторы и функцию distance. После чего выводим сначала первый результат(четные), затем второй(нечетные).

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_7.cpp>

# Задание 8

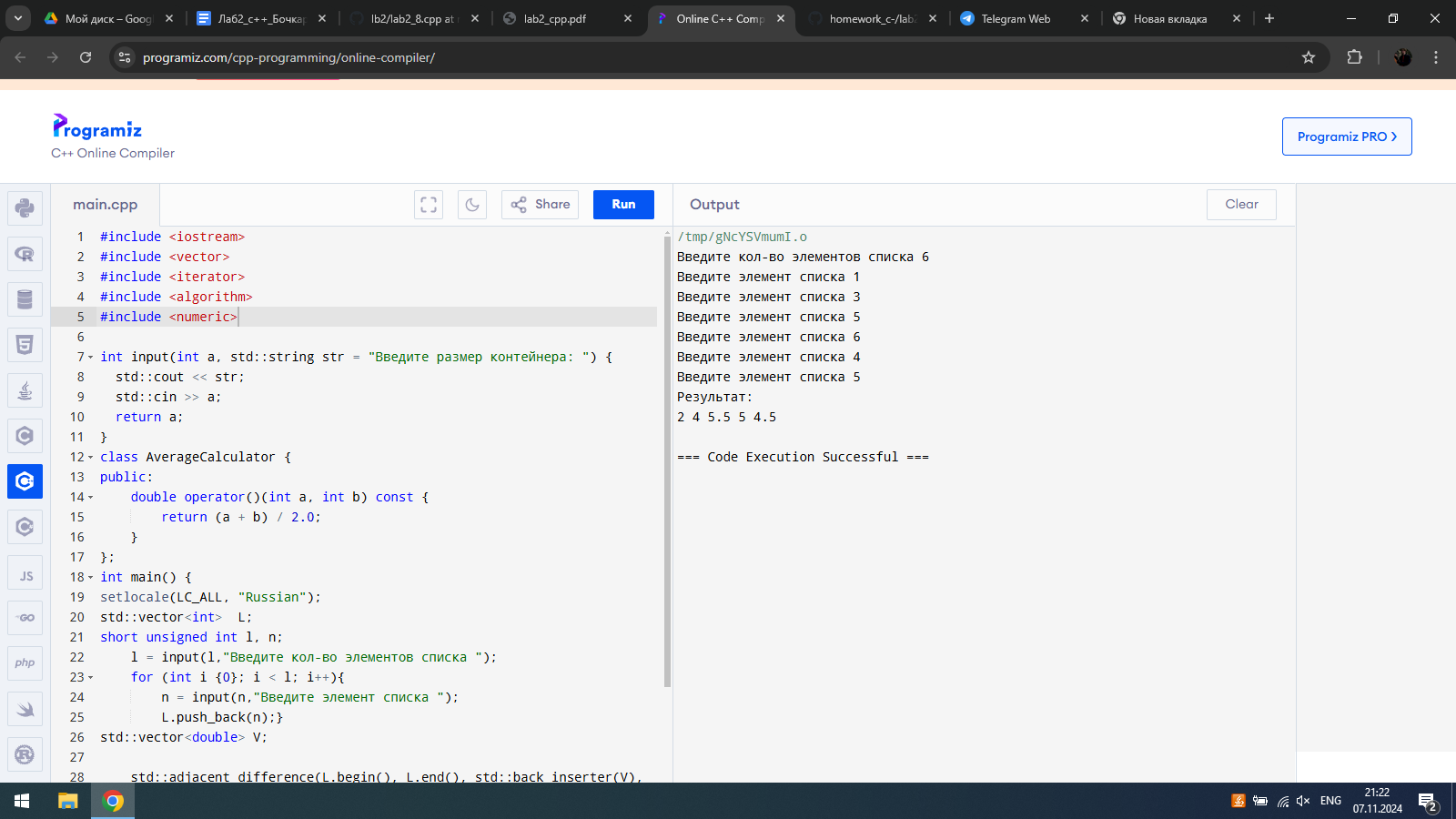
## Текст задания

Дан список L. Получить вектор V вещественных чисел, содержащий значения среднего арифметического для всех пар соседних элементов исходного списка (количество элементов вектора V должно быть на 1 меньше количества элементов списка L). Например, для исходного списка 1, 3, 4, 6 полученный вектор должен содержать значения 2.0, 3.5, 5.0. Использовать алгоритм adjacent\_difference с итератором вставки и функциональным объектом, а также функцию-член erase для вектора V.

## Алгоритм решения

Алгоритмом adjacent\_difference ищем среднее арифметическое для всех пар соседних элементов исходного списка. Удаляем первый элемент нашего вектора (т.к он равен первому элементу списка)

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_8.cpp>

# Задание 9

## Текст задания

Дан вектор V0, целое число N (> 0) и набор векторов V1, …, VN. Известно, что размер вектора V0 не превосходит размера любого из векторов V1, …, VN. Найти количество векторов VI, I = 1, …, N, в которых содержатся все элементы вектора V0 (без учета их повторений). Использовать алгоритм includes, применяя его в цикле к двум множествам, одно из которых создано на основе вектора V0, а другое на очередной итерации содержит элементы очередного из векторов VI, I = 1, …, N.

## Алгоритм решения

Перевести вектора в multiset, чтобы был учет повторений. Найти вхождения векторов Vi в вектор V0.

## Тестирование

## 

## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_9.cpp>

# Задание 10

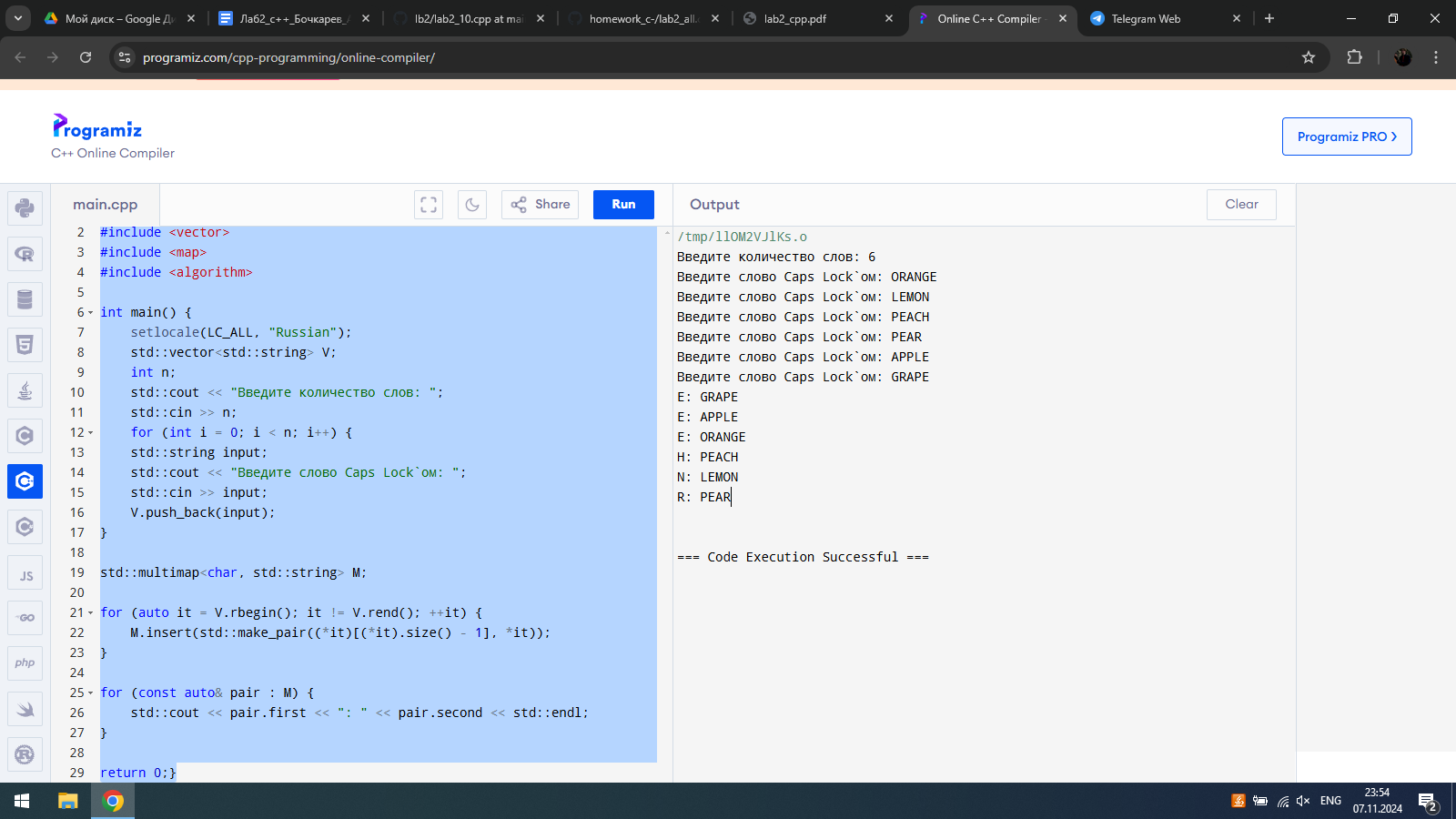
## Текст задания

Дан вектор V, элементами которого являются английские слова, набранные заглавными буквами. Выполнить группировку элементов вектора V, используя в качестве ключа группировки последнюю букву элемента: в одну группу должны входить все элементы вектора V, оканчивающиеся одной и той же буквой (сгруппированные элементы должны располагаться в порядке, обратном порядку их расположения в исходном векторе). Представить результат группировки в виде мультиотображения M, ключами которого являются ключи группировки, т. е. последние буквы элементов вектора V, а значениями — элементы вектора, оканчивающиеся на соответствующую букву (таким образом, отображение M должно иметь тип multimap). Вывести полученное отображение (для каждого элемента отображения M вначале выводить ключ, а затем связанный с ним элемент вектора V; ключи могут повторяться). Для перебора элементов контейнеров использовать цикл for по элементам контейнера.

## Алгоритм решения

Вводим количество слов и сами слова. Создаем multimap для отображения и создания пары. Перебираем и создаем пары из ключа и слова в цикле. Выводим получившиеся значения.

## Тестирование



## Код программы

<https://github.com/Mdcrych/lb2/blob/main/lab2_10.cpp>