**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**к курсовой работе**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2373 |  | Герасимов А |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2023

**Цель работы.**

Объединить все 4 лабораторные работы в единый проект. Нужно добавить инфраструктуру переключения между заданиями (интерактивное меню).

**Основные теоретические положения.**

Вы делаете объявления в файле заголовка, а затем используете директиву #include в каждом CPP-файле или другом файле заголовка, который требует этого объявления. Директива #include вставляет копию файла заголовка непосредственно в CPP-файл перед компиляцией.

Затем создайте файл реализации (обычно с расширением CPP или аналогичного расширения). Мы вызовем файл my\_class.cpp и предоставим определение для объявления члена. Мы добавим директиву #include для файла "my\_class.h", чтобы объявление my\_class вставлено в этот момент в CPP-файле, и мы добавим <iostream> в объявление .std::cout Обратите внимание, что кавычки используются для файлов заголовков в том же каталоге, что и исходный файл, а угловые скобки используются для заголовков стандартной библиотеки. Кроме того, многие заголовки стандартной библиотеки не имеют расширения H или других файлов.

Теперь мы можем использовать my\_class в другом CPP-файле. Мы #include файл заголовка, чтобы компилятор извлекал объявление.

**Постановка задачи.**

Объединить все 4 лабораторные работы в единый проект.

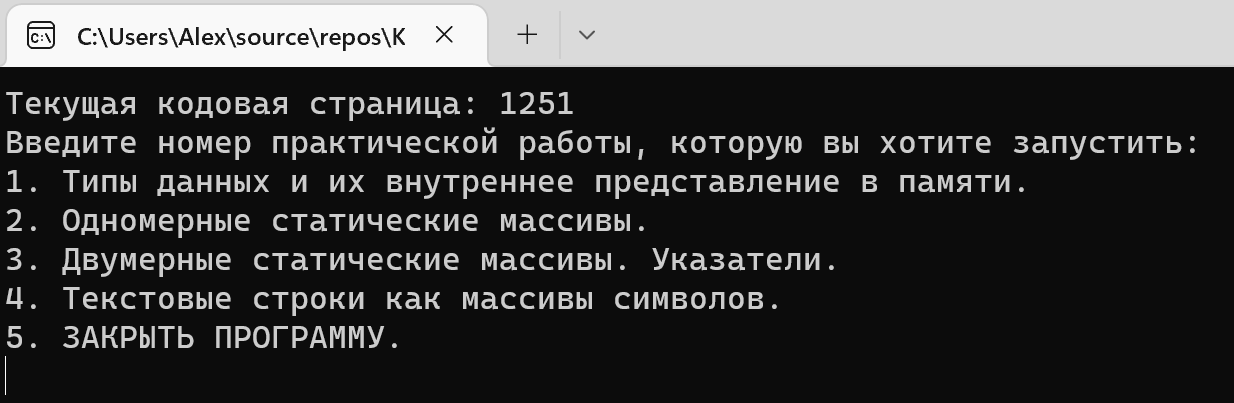
**Выполнение работы.**

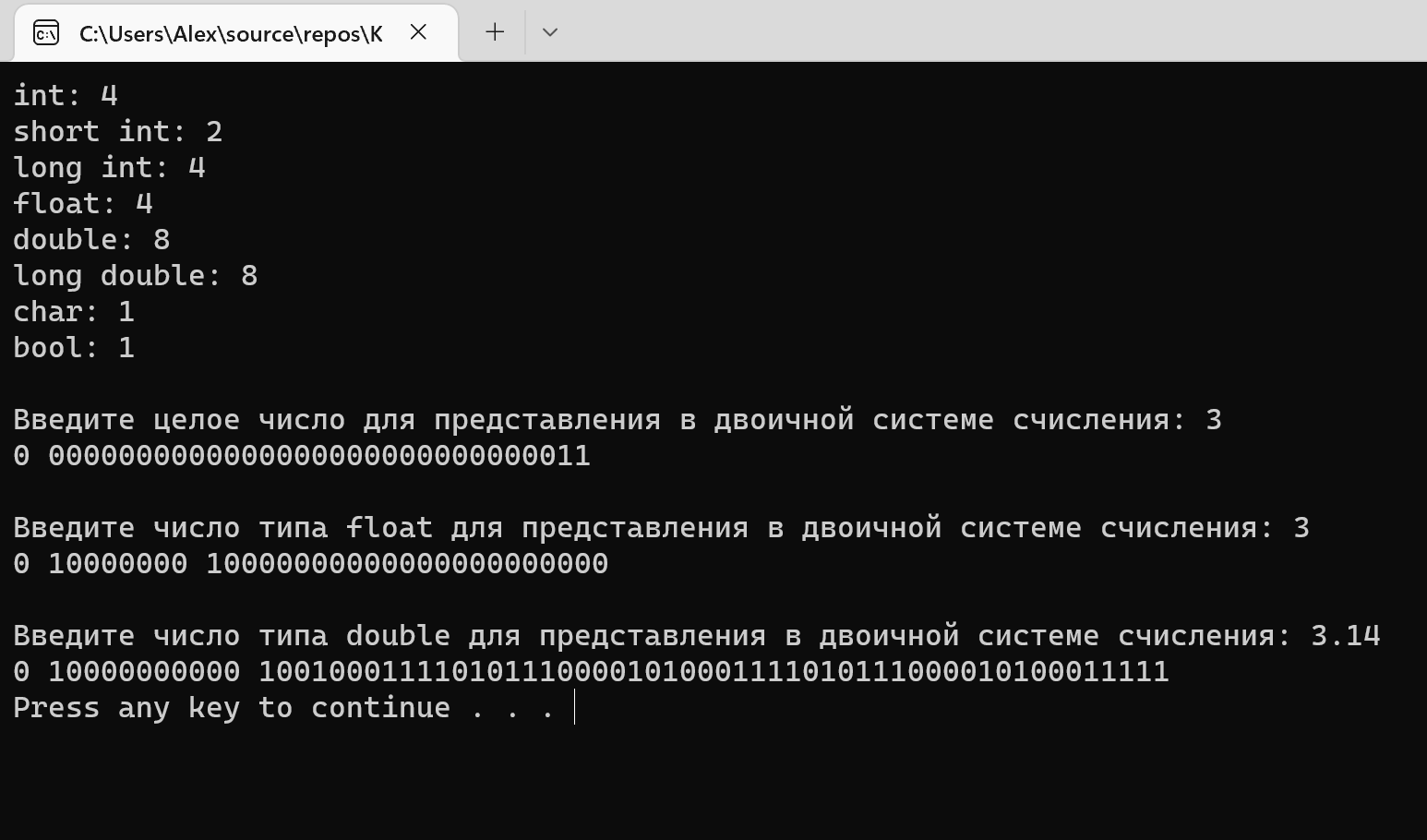
Код программы представлен на сайте github по ссылке: https://github.com/AleGers/Kursach.git

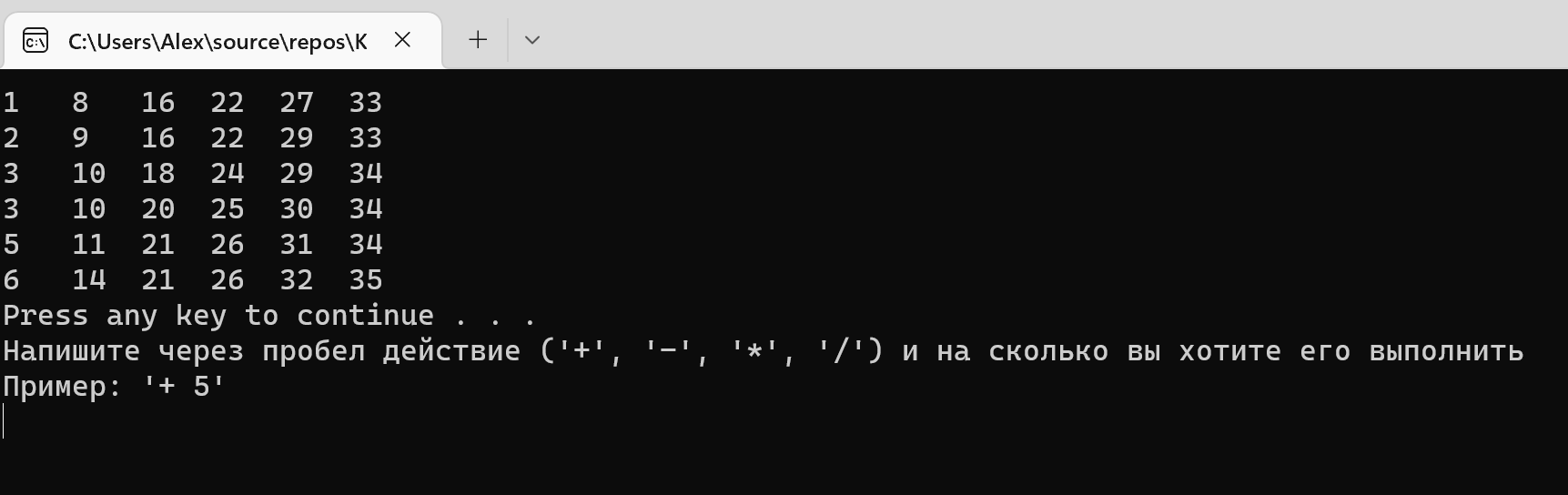
**Блок описания кода и использованных алгоритмов:**

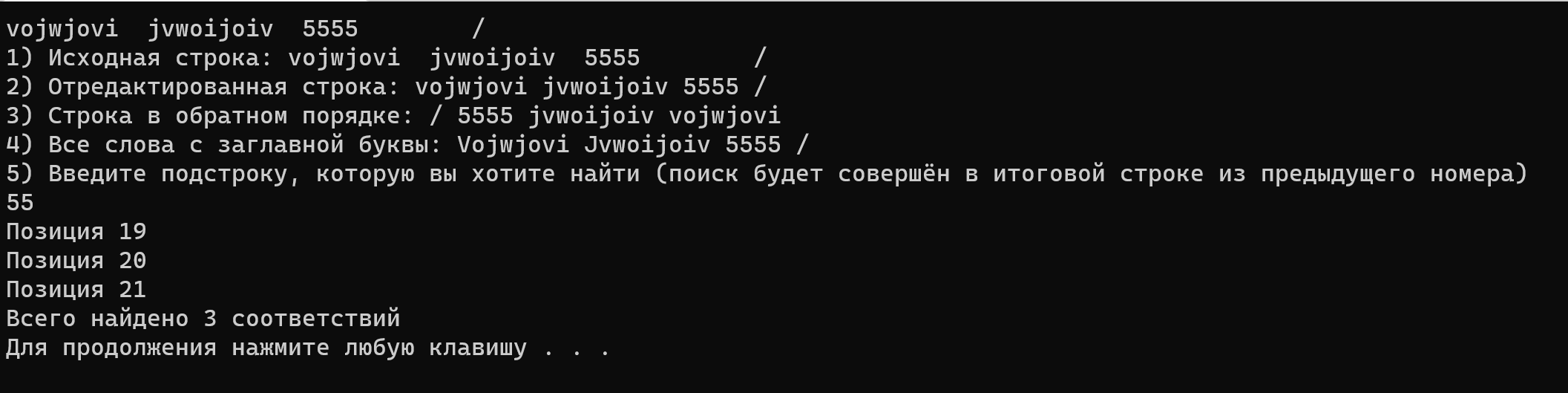
1. Создаём (.cpp) файл курсовой работы. В нём реализовываем меню с возможностью выбора практической работы. В каждом пункте идёт обращение к каждому отдельному файлу.
2. Создаём общий (.h) файл и объявляем функции, которые в дальнейшем будем прописывать в отдельных файлах.
3. Создаём четыре файла (.cpp). В каждом реализовываем практические работы. Во всех (.cpp) файлах прописываем «#include "Labs.h"», чтобы связать файлы между собой.

**Блок скриншотов работы программы**









**Выводы.**

Я объединил все 4 лабораторные работы в единый проект и добавил инфраструктуру переключения между заданиями. Научился работать с файлами.