

**Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«Ульяновский государственный университет»  
Факультет математики информационных и авиационных технологий**

**Итоговая лабораторная работа  
Шифр Цезаря для текстов**

**Выполнил: студент группы МОАИС-О-25/1**

**Григорьев Д. С.**

Ульяновск 2025

**Задача:** спроектировать и реализовать консольное приложение для шифрования текстов на C++, способное загружать, валидировать и сохранять данные в формате JSON.

**Среда выполнения:** Visual Studio 2022.

**Структура проекта:** Программа хранит тексты и их id в main.json. Шифрование текстов по id (пакетная обработка). Лог операций в log.json.

В main:

- id – id текста;
- content – сам текст;

В log:

- ts – изначальный текст;
- op – зашифрованный текст;
- key – ключ шифрования;
- id – id текста;

**Ключевые алгоритмы:**

**Шифрование:** функция «caesar» из заголовочного файла “Header.h” перебирает каждый символ строки в цикле. Если символ является буквой, то происходит сдвиг на key символов по алгоритму:  $\text{new\_text} += \text{base} + (\text{c} - \text{base} + \text{key}) \% 26$ , где:

- c – текущая буква;
- base – «a» либо «A», в зависимости от того, заглавная c или нет;
- key – на сколько символов сдвиг, параметр функции;
- new\_text – переменная, в которую записывается зашифрованный текст;

Иначе new\_text += c. Функция возвращает new\_text.

**Создание JSON:** За это отвечают функции create\_json и create\_log\_json из “Header.h”. Они создают json файл заданной структуры с помощью экранирования: \t, \n, \”. Если файл уже существует, и в него нужно дозаписать новые данные, то тогда:

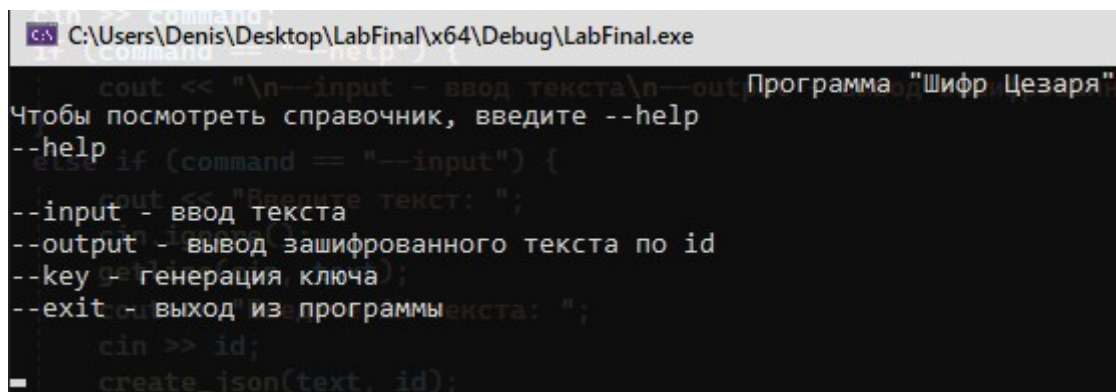
- создаётся новый файл с другим названием;
- в него записываются данные из старого, затем полученные данные;
- старый файл удаляется;
- новый переименовывается;

**Парсинг JSON:** Функция `read_json` из "Header.h" ищет в json файле `main` все тексты под заданным `id`, добавляет их в вектор в зашифрованном виде, затем возвращает вектор. Принцип работы:

- считывание каждой строки из `main` с помощью `while(getline)`;
- считывание строки до запятой с помощью `getline`;
- если подстрока, соответствующая `id`, равна параметру `id` функции, то считывается часть строки после запятой, и в вектор `v` добавляется зашифрованный текст, следующий после «"content":»;

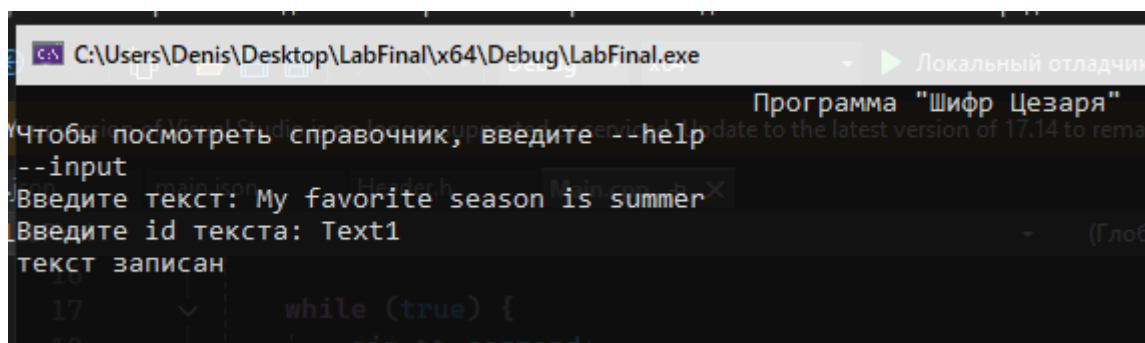
### Интерфейсы функций:

--help – справка о всех функциях



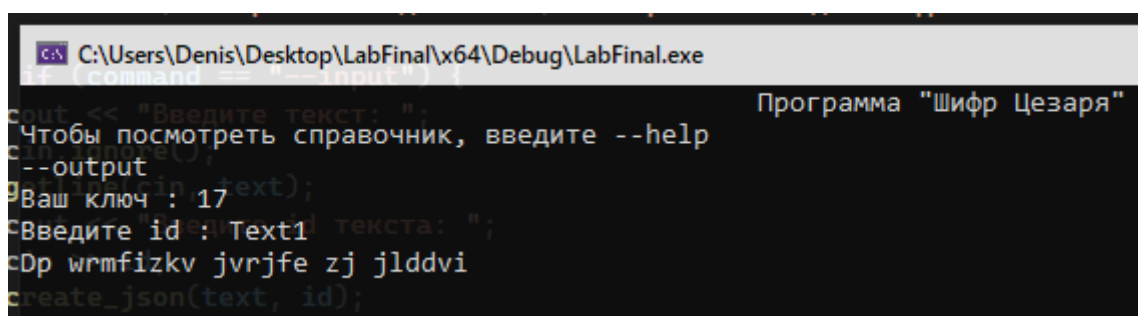
```
C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe
--help
cout << "\n--input - ввод текста\n--output - вывод зашифрованного текста по id\n--key - генерация ключа\n--exit - выход из программы\nЧтобы посмотреть справочник, введите --help\n--help\nelse if (command == "--input") {\n--input - ввод текста\n--output - вывод зашифрованного текста по id\n--key - генерация ключа\n--exit - выход из программы\n    cin >> id;\n    create_json(text, id);
```

--input – запись текста и его `id` в файл



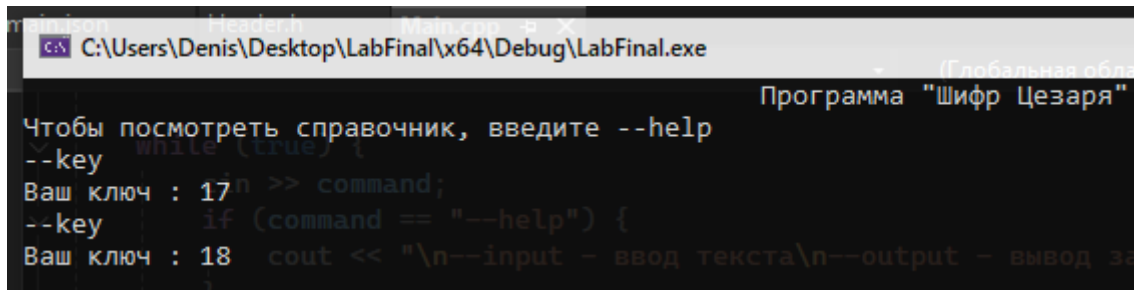
```
C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe
--input
Введите текст: My favorite season is summer
Введите id текста: Text1
текст записан
```

--output – вывод текста в зашифрованном виде (если ключ шифрования не был создан, программа генерирует его сама)



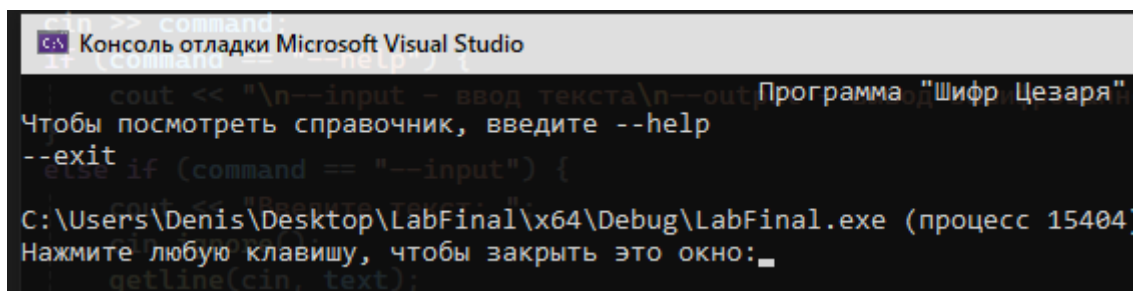
```
C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe
--output
Ваш ключ : 17
Введите id текста: Text1
Dp wrmfizkv jvrjfe zj jlddvi
create_json(text, id);
```

--key – создание ключа шифрования



```
C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe
Программа "Шифр Цезаря"
Чтобы посмотреть справочник, введите --help
--key
Ваш ключ : 17
--key
Ваш ключ : 18
cout << "\n--input - ввод текста\n--output - вывод эа
```

--exit – выход из программы



```
Консоль отладки Microsoft Visual Studio
--exit
cout << "\n--input - ввод текста\n--out
Программа "Шифр Цезаря"
Чтобы посмотреть справочник, введите --help
--exit
else if (command == "--input") {
C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe (процесс 15404)
Нажмите любую клавишу, чтобы закрыть это окно: _
```

### Обработка ошибок:

В случае, если пользователь введёт несуществующую команду, то выведется ошибка «Неизвестная команда».

### Результаты бенчмарков:

Количество символов	Время выполнения функции (сек)
50	0,0024517
100	0,0033258
1000	0,013389
2000	0,025389
5000	0,0507198

**Таким образом,** мне удалось реализовать программу, способную шифровать тексты и работать с файлами в json формате. Она обладает понятным интерфейсом и достаточной производительностью.