

**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 папке JSON\_Files json-файлах. Шифрование текстов по id (пакетная обработка). Лог операций в log.json.

В json-файлах:

- 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: `new_text += base + (c - base + key) % 26`, где:

- c – текущая буква;
- base – «а» либо «A», в зависимости от того, заглавная с 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_jsons` и `create_log_json` из “Header.h”. Они создают json файлы заданной структуры с помощью экранирования: \t, \n, \”. Количество создаваемых файлов задаётся переменной. Содержание файлов (текст и id) случайное.

**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
        cout << "\n--input - ввод текста\n--out Программа \"Шифр Цезаря\""
Чтобы посмотреть справочник, введите --help
--help
else if (command == "--input") {
--input - ввод текста
--output - вывод зашифрованного текста по id
--key Генерация ключа
--exit --выход из программы текста: ";
    cin >> id;
    create_json(text, id);
```

--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
        Программа "Шифр Цезаря"
Чтобы посмотреть справочник, введите --help
Введите команду: --output
Ваш ключ : 25
Введите название файла: file10.json
Ведите id : 58
Зашифрованные тексты: Интерфейсы функций:
mzmqpprbogndsejiiebymhwhrsmhnar
xtxrkhhxfervpzrbgmrerospofuqsqenipqhzxlanvszictyнкциях
iinldkuyysmhxohscouggjghtrdzxrkjpankeiyhrzpxlmkgdlakvvkjqnxkekisn
rafqwkxxiinbgujaoinxwzxlqqrsng
ixpjgunkcziogngwwmohqnijjkkmztxxzcfiiooqrqyvcrmkvqkkgzqwzktdrukth
```

--key – создание ключа шифрования

```
mainson Header.h Main.cpp ✘ X
cs C:\Users\Denis\Desktop\LabFinal\x64\Debug\LabFinal.exe
(Глобальная обработка)
Программа "Шифр Цезаря"

Чтобы посмотреть справочник, введите --help
--key
Ваш ключ : 17
--key      if (command == "--help") {
Ваш ключ : 18  cout << "\n--input - ввод текста\n--output - вывод з
    }
```

--exit – выход из программы

```
10 >> command;
cs Консоль отладки Microsoft Visual Studio
cout << "\n--input - ввод текста\n--output - вывод текста\nПрограмма "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:
    getline(cin, text);
```

### Обработка ошибок:

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

### Результаты бенчмарков:

Количество символов	Время выполнения функции (сек)
1000	0,000092
2000	0,000177
5000	0,000436
10000	0,000834
50000	0,004284
100000	0,0087338
250000	0,0214208
500000	0,0441346
1000000	0,0879955

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