Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет
имени первого Президента России Б.Н.Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Центр ускоренного обучения

РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ШИФРОФАНИЯ И ДЕШИФРОВАНИЯ ТЕКСТОВОГО ФАЙЛА МЕТОДОМ ВИЖЕНЕРА НА ЯЗЫКЕ C++

КУРСОВОЙ ПРОЕКТ

по дисциплине «Прикладное программирование»

Пояснительная записка
09.03.03 58.29.21 002 ПЗ

Ст. преподаватель:	О.Л. Чагаева
Нормоконтролёр:	О.Л. Чагаева
Студент группы РИЗ-200028у:	И.С. Арсентьев

Екатеринбург 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Человечество использует шифрование с того момента, как появилась первая секретная информация - такая, доступ к которой не должен быть публичным.

Шифрование появилось около четырех тысяч лет тому назад. Первым известным примером шифра считается египетский текст, созданный примерно в 1900 г. до н. э., в котором вместо обычных для египтян иероглифов использовались не совпадающие с ними символы.

Вообще, историю криптографии можно считать равной по возрасту истории существования письменности, потому что именно с появлением письменности возникла потребность придумывать различные способы для хранения информации в виде, доступном только для определенного круга лиц. Например, до нашей эры был придуман известный «Шифр Цезаря», который заключался в замене каждого символа в тексте на элемент, отстоящий от него в алфавите на фиксированное число позиций. Естественно, что люди, от которых информация утаивалась, искали всяческие способы расшифровать закодированные сообщения. Таких людей сейчас называют криптоаналитиками. Обе враждующие стороны находились в постоянном противоборстве: первые постоянно придумывали новые шифры, с первого взгляда недоступные для криптоаналитиков, а вторые находили способы дешифровки скрытых сообщений.

Благодаря работе Абу аль – Кинди ((около 801—873)- арабский математик) оказалось, что шифры типа «Шифра Цезаря» (то есть моноалфавитные шифры, в которых каждой букве кодируемого текста ставится в соответствие однозначно какая-то шифрованная буква) довольно-таки легко поддаются частотному криптоанализу. Возникла потребность в разработке таких шифров, ручная расшифровка которых может потребовать очень значительных усилий. И на смену моноалфавитным шифрам пришли полиалфавитные шифры. Абу аль – Кинди первым предложил использовать многоалфавитный шифр. В европейских странах это произошло в эпоху Возрождения, когда развитие торговли потребовало надежные способы защиты информации. Одним из первых предложил полиалфавитный шифр итальянский архитектор Батисте Альберти. Впоследствии данный шифр получил имя дипломата XVI века Блеза де Виженера. Также вклад в развитие полиалфавитных шифров внес немецкий аббат XVI века Иоганн Трисемус. Простым, но стойким способом полиалфавитной замены является шифр Плейфера, открытый в начале XIX века Чарльзом Уитстоном.

Этот шифр использовался вплоть до I мировой войны. Последним словом в развитии полиалфавитных шифров стали так называемые роторные машины, которые позволяли легко создавать устойчивые к криптоатакам полиалфавитные шифры. Примером такой

машины является немецкая машина Enigma, разработанная в 1917 г. Эдвардом Хеберном.

Суть шифрования заключается в предотвращении просмотра исходного содержания сообщения теми, у кого нет средств его дешифровки.

Шифрование — это способ сокрытия исходного смысла сообщения или другого документа, обеспечивающий искажение его первоначального содержимого. Зачастую в более сложных шифрах для расшифровки уже зашифрованного сообщения помимо знания правил шифрования, требуется ключ к шифру. Под ключом в данном случае подразумевается конкретное секретное состояние параметров алгоритмов шифрования и дешифрования.

Алфавит - законченное множество используемых для кодирования информации символов. Текстом будем понимать упорядоченную последовательность из символов алфавита.

Суть шифрования заключается в том, чтобы скрыть информацию от тех, для кого она не предназначается, даже если они могут видеть сам зашифрованный текст. Противоположный процесс по обращению зашифрованного текста в его оригинальный вид называется дешифрованием.

Актуальность исследования данной курсовой работы обосновывается необходимостью создания консольного приложения на языке C++ для демонстрации процесса шифрования исходного текста в закодированный с помощью ключа, затем для процесса дешифрования.

В качестве языка программирования для создания данного приложения используется объектно-ориентированный язык C++.

Объектом исследования в курсовой работе является процесс шифрования исходного текста на английском языке, использование ключа (слова на английском языке), процесса дешифрования зашифрованного файла в читаемый текст, имеющий логический смысл. Подразумеваем, что входные данные (исходный файл для кодирования), выходные данные (файл после дешифрования), а также ключ (файл, содержащий кодовое слово) находятся в оговорённом каталоге на жёстком диске компьютера.

Предмет исследования представляет собой реализацию проекта на языке программирования C++, содержащего в себе компоненты для полноценной работы консольного приложения, демонстрирующего возможность шифрования и дешифрования текста методом Виженера.

Задачей исследования в курсовой работе является написание программы на языке высокого уровня C++, которая позволит выполнять шифрование текста в файле с помощью ключа, а также выполнять обратную операцию дешифрования.

Проект носит ознакомительный учебный характер, демонстрирующий работу алгоритма шифрования методом Виженера.

В курсовом проекте имеется историческая справка о различных методах шифрования, подробно описан процесс шифрования текста методом Виженера, алгоритм шифрования и дешифрования реализован для консольного приложения на языке C++, отладка и запуск написанного кода в среде разработки MS Visual Studio 2019.

Курсовой проект содержит введение, теоретическую часть, в которую включена историческая справка и постановка задачи для проекта, практическую часть, в которой описан процесс реализации алгоритма шифрования и дешифрования файла, заключение с подведением итогов, приложения с вложениями, иллюстрирующими работу написанной программы в компиляторе.

1 Постановка задачи для реализации проекта

1.1 Историческая справка о различных методах шифрования данных

Исторически сложилось множество различных способов шифрования текста и данных у различных народов и цивилизаций. Приведём некоторые наиболее известные, которые представляют академический интерес для изучения и понимания общего процесса работы с современными шифрами.

Одним из самых простых алгоритмов шифрования является атбаш. Атба́ш (ивр. אתב"ש) — простой шифр подстановки для иврита. Правило шифрования состоит в замене i-й буквы алфавита буквой с номером n − i + 1, где n — число букв в алфавите.

Пример для латинского алфавита выглядит как:

Исходный текст: abcdefghijklmnopqrstuvwxyz

Зашифрованный текст: ZYXWVUTSRQPONMLKJIHGFEDCBA

Происхождение слова «атбаш» объясняется принципом замены букв. Слово אתבש составлено из букв «алеф», «тав», «бет» и «шин», то есть первой и последней, второй и предпоследней букв еврейского алфавита.

Шифр атбаш был изобретен Ессеями - иудейской сектой повстанцев. Они разработали множество различных кодов и шифров, которые использовались для сокрытия важных имен и названий, чтобы потом избежать преследования. Знания этих кодов и шифров были потом переданы Гностикам, которые, в свою очередь, передали их Катарам. Позже Орден Тамплиеров завербовал Катарских дворян и перенял знания шифров. Таким образом, шифр был использован на протяжении многих лет, от около 500 до н.э. до 1300 г. н.э. — момента, когда Орден Тамплиеров был распущен.

В античном мире в эпоху римской империи имел широкое распространение шифр Цезаря (шифр сдвига) — один из самых простых и наиболее широко известных методов шифрования. Правило шифрования реализует кодирование фразы путем «сдвига» всех букв фразы на определенное число n (в оригинальном шифре Цезаря число n равнялось 3). Если буква кодируемой фразы имеет в алфавите позицию j, то она в "шифровке" будет заменяться буквой, находящейся в алфавите на позиции j + n.

Пример для русского алфавита выглядит так:

Исходный текст содержит фразу: Съешь же ещё этих мягких французских булок, да выпей чаю.

Зашифрованный текст шифром Цезаря будет иметь вид: Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб.

Согласно «Жизни двенадцати цезарей» Светония, в I в до н. э. Гай Юлий Цезарь во время войны с галлами, переписываясь со своими генералами в Риме, заменял в сообщении

первую букву латинского алфавита (A) на четвертую (D), вторую (B) – на пятую (E), наконец, последнюю – на третью. Цезарь использовал сдвиг на три позиции.

Не менее интересным методом античного шифрования текстов является квадрат Полибия (англ. Polybius square), также известный как шахматная доска Полибия — оригинальный код простой замены. Правило шифрования наиболее интересно применяется к латинскому алфавиту

Применительно к современному латинскому алфавиту из 26 букв шифрование по этому квадрату заключалось в следующем. В квадрат размером 5x6 клеток выписываются все буквы алфавита, при этом буквы I, J не различаются (J отождествляется с буквой I). Шифруемая буква заменялась на координаты квадрата, в котором она записана. Так, B заменялась на AB, F на BA, R на DB и т.д. При расшифровании каждая такая пара определяла соответствующую букву сообщения. Ключом такого шифра являлось расположение букв в таблице, к примеру 5*5. Начальное расположение букв должно определяться ключом. В современном латинском алфавите 26 букв, следовательно таблица должна состоять из 5 строк и 5 столбцов, так как 25=5*5 наиболее близкое к 26 число. Но так как английский алфавит всё же содержит 26 букв, мной будет разработана программа из 6 строк и 6 столбцов, содержащая все символы латинского алфавита.

Пример для латинского алфавита выглядит так:

Исходный текст: abcdefghijklmnopqrstuvwxyz

Зашифрованный текст:

DCDDDEDFDGDHEAEBECEDEEEFEGEHFAFBFCDGFEFFFGFHGAGBGCGD

Данный вид кодирования изначально применялся для греческого алфавита, но затем был распространен на другие языки. Квадрат Полибия - одна из древнейших систем кодирования, был разработан Полибием (греческий историк, полководец, государственный деятель, III век до н. э.).

С развитием шифровального дела в новом времени, в европейской части мира был разработан шифр Гронсфельда - многоалфавитный шифр сложной замены.

Схема очень напоминает шифр Цезаря, дополненного числовым ключом. При шифровании сообщения его выписывают в строку, а под ним ключ. Если ключ оказался короче сообщения, то его циклически повторяют. Шифровку получают, находя символ в колонке таблицы по букве текста и строке, соответствующей букве ключа.

Для примера, пусть нам надо зашифровать слово EXALTATION. Например, берём в качестве ключа число 31415, затем составляем следующую таблицу:

E	X	A	L	T	A	T	I	O	N
3	1	4	1	5	3	1	4	1	5

Рисунок 1- Таблица Гронсфельда с ключом

Получается, что каждой букве соответствует некая цифра, это цифра будет показывать, на сколько позиций будет происходить смещение алфавита для каждой конкретной буквы. Например, покажем, как преобразуется буква Е:

A	B	C	D	E	F	G	H
					1	2	3

Рисунок 2- Таблица Гронсфельда. Смещение символа алфавита

То есть букве Е соответствует буква Н. Таким образом, для всего слова получаем зашифрованный текст: HYEMYDUMPS. Обратное преобразование происходит подобным образом, только каждый раз сдвигаем алфавит в другую сторону.

Исходный текст содержит фразу: EXALTATION

Зашифрованный текст получим как: HYEMYDUMPS

В качестве ключа используем комбинацию чисел 31415

Идея использования лозунга (гаммы) без изображения указанных выше таблиц была предложена в 1734 году бельгийцем Хосе де Бронкхором и начальником первого дешифровального отделения в Германии, военным и дипломатом графом Гронсфельдом.

Наиболее интересным для изучения и демонстрации простого шифрования современными программными средствами является шифр Виженера

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Является простой формой многоалфавитной замены.

Первое точное документированное описание многоалфавитного шифра было сформулированно Леоном Баттиста Альберти в 1467 году, для переключения между алфавитами использовался металлический шифровальный диск. Система Альберти переключает алфавиты после нескольких зашифрованных слов. Позднее, в 1518 году, Иоганн Трисемус в своей работе «Полиграфия» изобрел tabula recta — центральный компонент шифра Виженера.

То, что сейчас известно под шифром Виженера, впервые описал Джованни Батиста Беллазо в своей книге La cifra del. Sig. Giovan Battista Bellaso. Он использовал идею tabula recta Трисемуса, но добавил ключ для переключения алфавитов шифра через каждую букву.

Состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На разных этапах кодировки шифр Виженера использует различные алфавиты из этой таблицы. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рисунок 3- Таблица Виженера (tabula recta)

Например, предположим, что исходный текст имеет вид: ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: LEMONLEMONLE

Стоит учитывать, что чем объёмнее будет текст, тем значительнее будет ключ для шифрования, поэтому для практического применения в настоящее время требуется сложный ключ.

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Зашифрованный текст: LXFOPVEFRNHR

Ключ: LEMONLEMONLE

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы A-Z соответствуют числам от 0 до 25, то шифрование Виженера можно записать в виде формулы:

$$C_i \equiv (P_i + K_i) \mod 26$$

Рисунок 4- Формула для шифрования методом Виженера

$$P_i \equiv (C_i - K_i + 26) \mod 26$$

Рисунок 5- Формула для дешифрования методом Виженера

2 Создание программного продукта

2.1 Определение схемы работы программы шифрования файла

Для понимания процесса работы программы шифрования текста из файла Input.txt при помощи файла с ключом Key.txt стоит привести блок-схему процесса, после которой будет проведено разъяснение работы отдельно взятых элементов схемы.

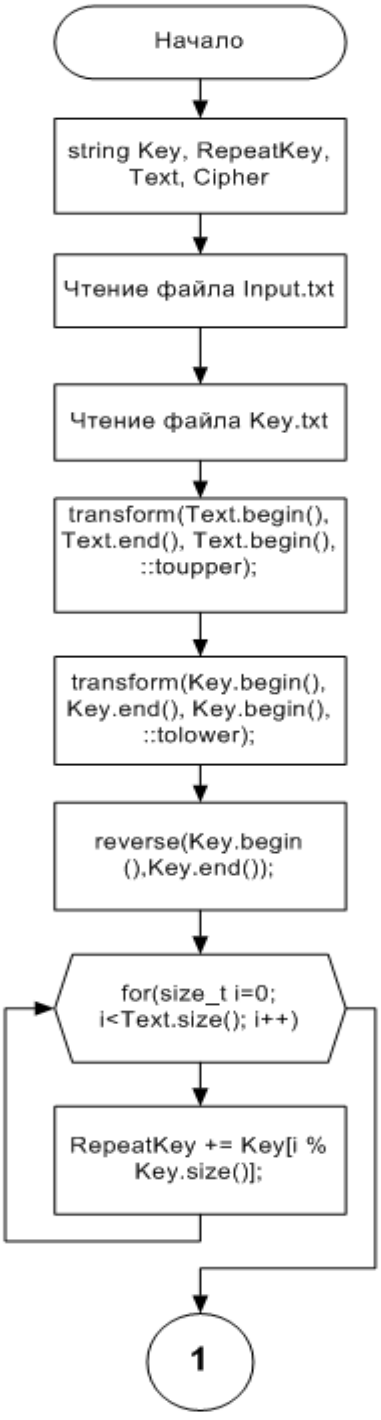


Рисунок 4- Блок-схема работы программы шифрования

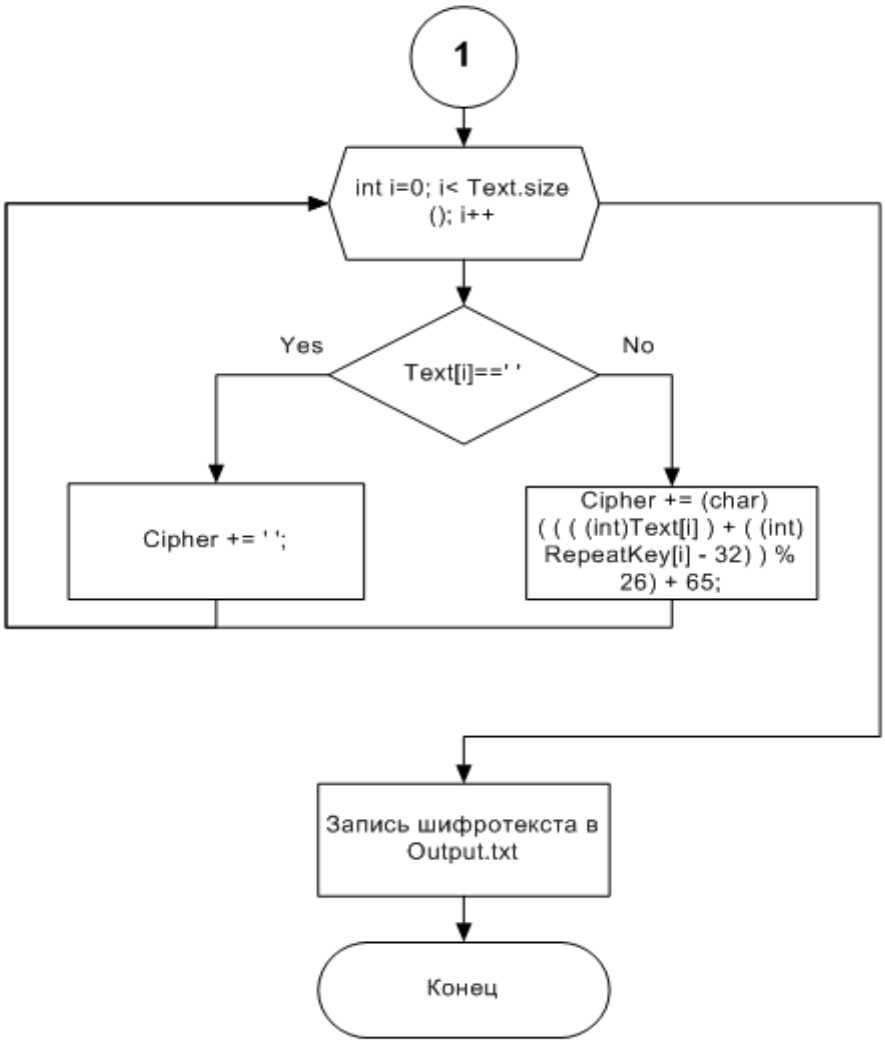


Рисунок 5- Блок-схема работы программы шифрования(продолжение)

Программа на входе будет принимать значение из файла Input.txt. Путём преобразования через файл ключа Key.txt, будет сформирован поток данных, который будет записан в файл Output.txt. Предполагаем, что все файлы, с которыми будет работать программа шифрования, лежат в одной папке.

Разработка приложения будет производиться в среде разработки Microsoft Visual Studio 2022 для архитектуры x64 с операционной системой Windows 10 Professional.

Для корректной работы механизма программы, необходимо подключение необходимых библиотек. Размещаем их перед телом программы:

Листинг 1. Подключение библиотек в программу шифрования

```
#include <iostream>
#include <conio.h>
#include <algorithm>
```

```
#include <fstream>
     #include <string>
```

Будем использовать пространство имён using namespace std.

Для отображения текста в окне командной строки в тело основной программы включаем setlocale(LC_ALL, "Russian").

Введём переменные Key, RepeatKey, Text, Cipher строкового типа, с которыми будет работать программа.

Путём передачи содержимого входного файла в переменную Text, переменная будет выведена на терминал в командной строке.

Для исключения неправильного отображения символов в исходном сообщении необходимо изменить регистр символов.

Ниже будет выведена строка с содержащимся ключом для шифрования из фала Key.txt.

После вывода текста ключа необходимо инвертировать символы, расположив в строке символы задом наперёд, от последнего к начальному.

Квадрат Виженера будет заполнен повтором ключа до тех пор, пока количество символов ключа не составит длину исходного файла.

Таким образом, файл посимвольно шифруется при помощи ключа – ключевого слова в файле Key.txt.

Иначе говоря, массив символов из исходного файла сопоставляется с массивом символов ключа и на пересечении символов выбирается шифрующий символ, выполняется целочисленное деление на количество символов используемого алфавита.

Фрагмент кода программы, который иллюстрирует работу механизма шифрования приведён на листинге 2:

Листинг 2. Фрагмент кода, иллюстрирующий процесс шифрования текст

```
//Шифрование исходного текста
//по формуле: C = T + K (mod 26) , где C - зашифрованный символ,
//T - символ исходного текста, K - символ ключа
for (int i = 0; i < Text.size(); i++)
     {
          if (Text[i] == ' ')
          {
               Cipher += ' ';
          }
          else
          {
Cipher += (char)((((int)Text[i]) + ((int)RepeatKey[i] - 32)) %
26) + 65;
     }
}
cout << "Зашифрованный текст: " << endl << Cipher << endl;
```

После выполнения шифрования полученный массив передаётся потоком в текстовый файл Output.txt, в который и записывается. После записи текста в файл, текст отображается в окне командной строки, таким образом демонстрируя пользователю результат выполнения кода в программе.

Для нашего эксперимента выберем фрагмент текста на английском языке без знаков препинания:

The artist is the creator of beautiful things To reveal art and conceal the artist is arts aim The critic is he who can translate into another manner or a new material his impression of beautiful things The highest as the lowest form of criticism is a mode of autobiography Those who find ugly meanings in beautiful things are corrupt without being charming This is a fault Those who find beautiful meanings in beautiful things are the cultivated For these there is hope They are the elect to whom beautiful things mean only beauty There is no such thing as a moral or an immoral book Books are well written or badly written That is all The nineteenth century dislike of realism is the rage of Caliban seeing his own face in a glass

В качестве ключа запишем в файл Key.txt слово VIGENER.

После процесса шифрования получим зашифрованный текст вида:

XYI EXBDWK VW BCI GEIGBJV SS HMVYKMSYR OLZRTW BJ IIIIGT EIX ETL GFRPIGT XYI EXBDWK VW IMXJ NMS OLV PVOBDG MF NM AYS GGV XIEAWRIOI MAXU VRFXUIX HEERRV WM R AIC HEKIEMGT LZW MSXMIJWVST JJ FREABDJLP XNQIKJ GLK CMXLRWZ VW XUI TJAVWG LWMQ SS IZDXZGVWS DW E QULZ FJ EABJFZSTVGXCC XUSYM AYS JOVY LKYC UZEEMAKY DR FREABDJLP XNQIKJ NVK XSIVHTZ RMKLBYZ WIZRT IPVVDMAK BCMJ VW I JRYYX BCSJI ANW JZRQ HMVYKMSYR HIRRVRMA ME OIGCOMWYY ZPDRXW EXM XYI GATOMMEGIJ ASI GLKAZ KLRVK DW LBTK OLVC EXM XYI IRMXX XB CPJQ FREABDJLP XNQIKJ ZIGV SEPL HMVYKC XNMMI MF TW WLGU ZPDRX NW I QFVNP WM RR MSUJVRP FUWFL FBSQA EII AKTG NVVXZMI FV FGLGC AEMZBZR XUEZ DW EYP BCI RVRKBZIEXU IMIXLVL JQNPZOR UN VVEYMYU MJ GLK MEXI SL XECMOET NIVMAK PDW SJR NVGV VR I KCEFW

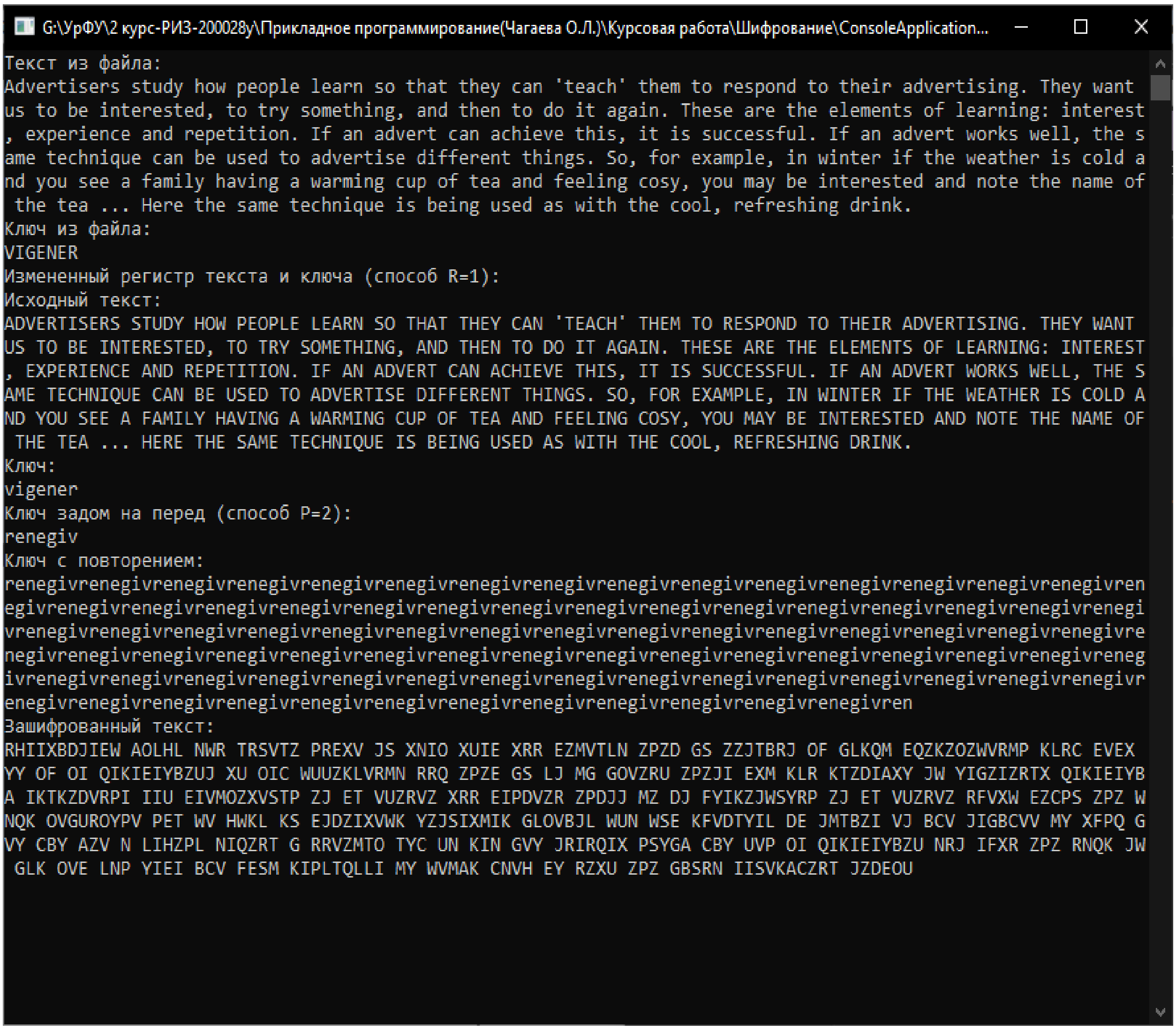
После вывода шифротекста в командной строке, будет произведена запись в файл Output.txt, находящийся в папке с исходным текстом и ключом.

Запись текста в файл производится также потоком.

После получения зашифрованного файла работа программы останавливается. Полученный файл не является осмысленным без применения процедуры дешифровки.

Стоит привести программу, являющейся обратной к разработанной, которая позволит перевести зашифрованный файл в читаемый при помощи исходного ключа.

Результат работы программы шифрования файла иллюстрирует рисунок 6.



```
G:\УрФУ\2 курс-РИЗ-200028у\Прикладное программирование(Чагаева О.Л.)\Курсовая работа\Шифрование\ConsoleApplication...
Текст из файла:
Advertisers study how people learn so that they can 'teach' them to respond to their advertising. They want
us to be interested, to try something, and then to do it again. These are the elements of learning: interest
, experience and repetition. If an advert can achieve this, it is successful. If an advert works well, the s
ame technique can be used to advertise different things. So, for example, in winter if the weather is cold a
nd you see a family having a warming cup of tea and feeling cosy, you may be interested and note the name of
 the tea ... Here the same technique is being used as with the cool, refreshing drink.
Ключ из файла:
VIGENER
Измененный регистр текста и ключа (способ R=1):
Исходный текст:
ADVERTISERS STUDY HOW PEOPLE LEARN SO THAT THEY CAN 'TEACH' THEM TO RESPOND TO THEIR ADVERTISING. THEY WANT
US TO BE INTERESTED, TO TRY SOMETHING, AND THEN TO DO IT AGAIN. THESE ARE THE ELEMENTS OF LEARNING: INTEREST
, EXPERIENCE AND REPETITION. IF AN ADVERT CAN ACHIEVE THIS, IT IS SUCCESSFUL. IF AN ADVERT WORKS WELL, THE S
AME TECHNIQUE CAN BE USED TO ADVERTISE DIFFERENT THINGS. SO, FOR EXAMPLE, IN WINTER IF THE WEATHER IS COLD A
ND YOU SEE A FAMILY HAVING A WARMING CUP OF TEA AND FEELING COSY, YOU MAY BE INTERESTED AND NOTE THE NAME OF
 THE TEA ... HERE THE SAME TECHNIQUE IS BEING USED AS WITH THE COOL, REFRESHING DRINK.
Ключ:
vigener
Ключ задом на перед (способ P=2):
renegiv
Ключ с повторением:
renegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivren
egivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegi
vrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivre
negivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivreneg
ivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivr
enegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivrenegivren
Зашифрованный текст:
RHIIXBDJIEW AOLHL NWR TRSVTZ PREXV JS XNIO XUIE XRR EZMVTLN ZPZD GS ZZJTBRJ OF GLKQM EQZKZOZWVRMP KLRC EVEX
YY OF OI QIKIEIYBZUJ XU OIC WUUZKLVRMN RRQ ZPZE GS LJ MG GOVZRU ZPZJI EXM KLR KTZDIAXY JW YIGZIZRTX QIKIEIYB
A IKTKZDVRPI IIU EIVMOZXVSTP ZJ ET VUZRVZ XRR EIPDVZR ZPDJJ MZ DJ FYIKZJWSYRP ZJ ET VUZRVZ RFVXW EZCPS ZPZ W
NQK OVGUROYPV PET WV HWKL KS EJDZIXVWK YZJSIXMIK GLOVBJL WUN WSE KFVDTYIL DE JMTBZI VJ BCV JIGBCVV MY XFPQ G
VY CBY AZV N LIHZPL NIQZRT G RRVZMTO TYC UN KIN GVY JRIRQIX PSYGA CBY UVP OI QIKIEIYBZU NRJ IFXR ZPZ RNQK JW
 GLK OVE LNP YIEI BCV FESM KIPLTQLLI MY WVMAK CNVH EY RZXU ZPZ GBSRN IISVKACZRT JZDEOU
```

Рисунок 6- Результат работы программы шифрования

2.1 Определение схемы работы программы дешифрования файла

Для понимания процесса работы программы дешифрования текста из файла Output.txt при помощи файла с ключом deKey.txt стоит привести блок-схему процесса, после которой будет проведено разъяснение работы отдельно взятых элементов схемы.

Аналогично первой программе, для процесса дешифрования файла будут использоваться потоки данных, в которых будет посимвольно передаваться зашифрованный текст. Для расшифровки будем использовать текст из файла deKey.txt, длину которого будем увеличивать до требуемой величины исходного шифротекста.

Как и в предыдущей программе для шифрования будет подключён модуль необходимых библиотек для работы, будет использоваться то же пространство имён и аналогичная работа с потоками.

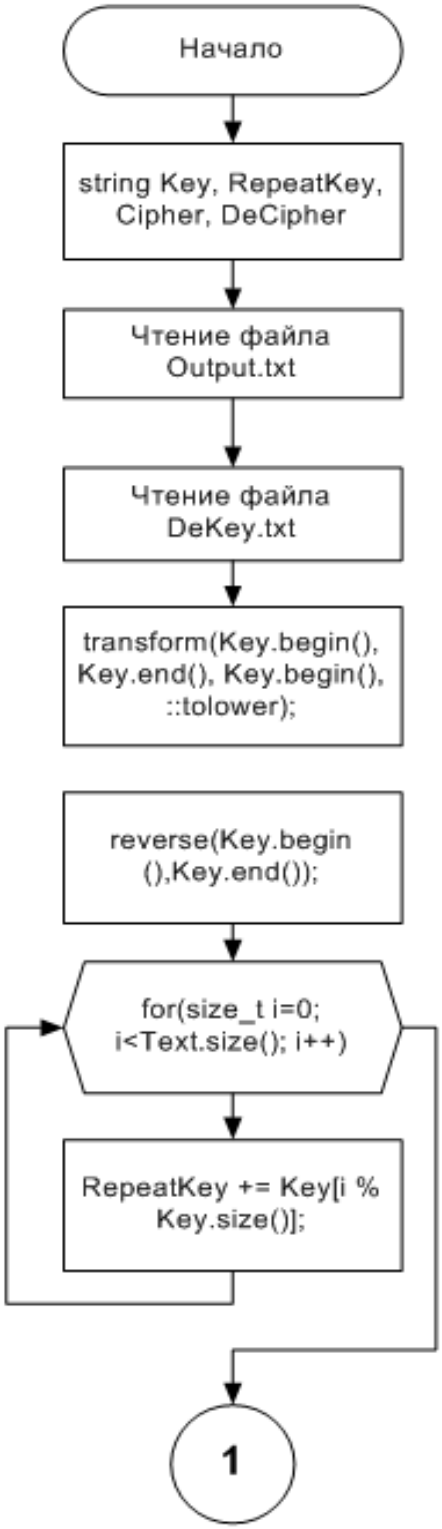


Рисунок 7- Блок-схема работы программы дешифрования

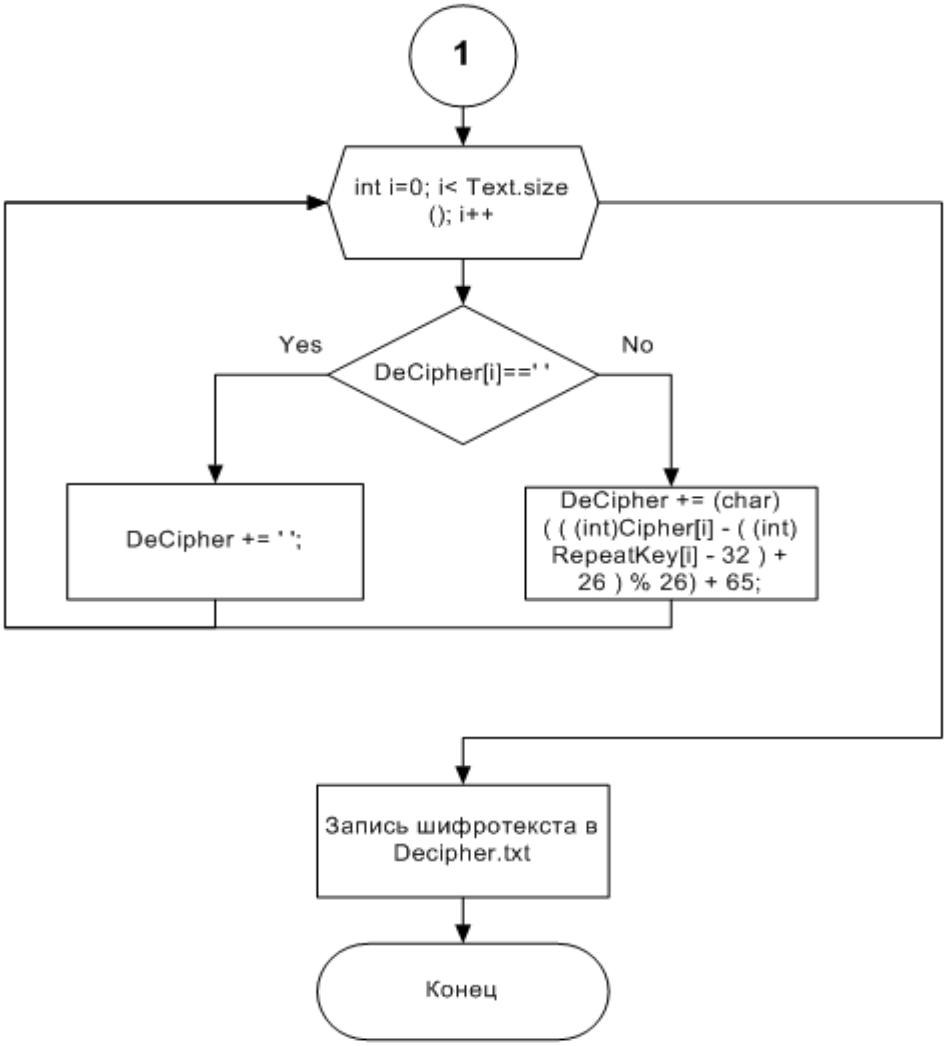


Рисунок 8- Блок-схема работы программы дешифрования(продолжение)

Отличие лишь имеется в механизме дешифрования шифротекста в осмысленный файл. Реализация механизма дешифрования изложена в листинге 3:

Листинг 3. Фрагмент кода, иллюстрирующий процесс дешифрования шифротекста

```
//Расшифровка зашифрованного текста
     //по формуле: T = C - K + 26 (mod 26) , где C -
//зашифрованный символ,T - символ исходного текста, K - символ
//ключа
for (int i = 0; i < Cipher.size(); i++)
     {
     if (Cipher[i] == ' ')
     {
     DeCipher += ' ';
     }
     else
     {
```

```
DeCipher += (char)(((int)Cipher[i] - ((int)RepeatKey[i] - 32) +
26) % 26) + 65;
     }
}
          cout << "Расшифрованный текст: " << endl << DeCipher <<
endl;
```

Таким образом, программа получает на вход поток символов из зашифрованного файла. Путём преобразований с помощью квадрата Виженера, составленного из размноженного значения ключа, на выходе программы формируется поток, записываемый в файл Decipher.txt.

Все действия программы также отображаются в терминале командной строки. Если мы будем использовать зашифрованный текст и ключ из предыдущего примера с шифрованием, в окне программы увидим результат:

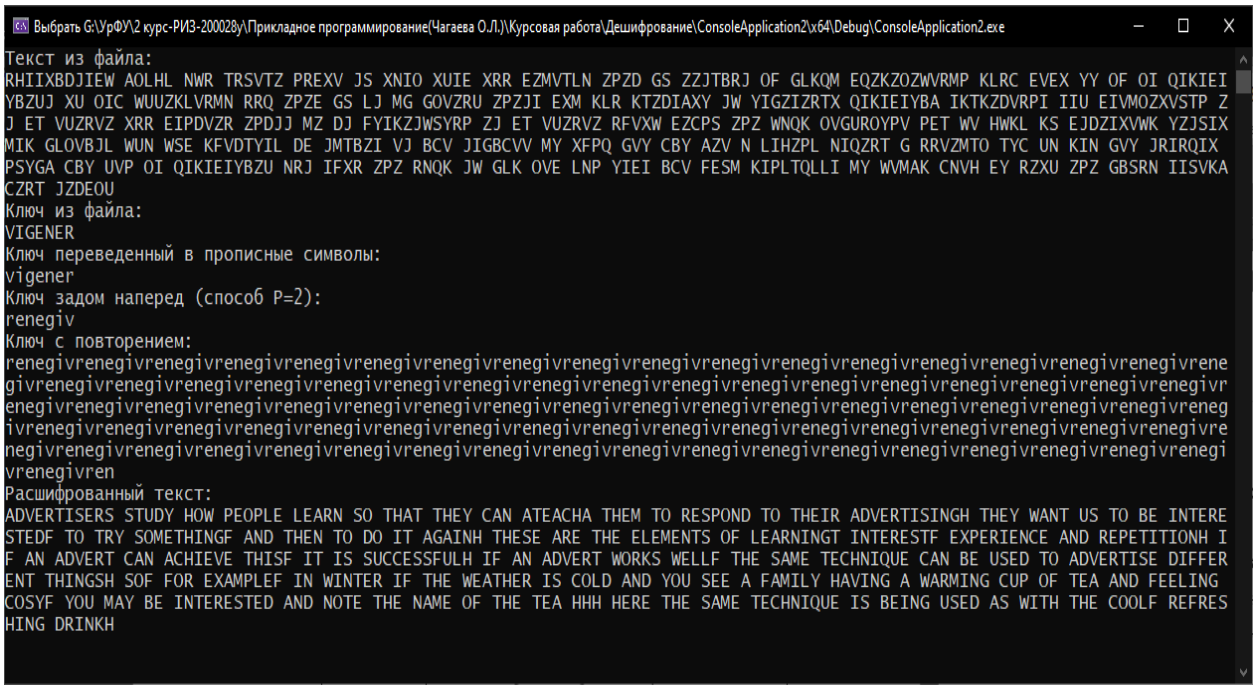


Рисунок 9- Результат работы программы дешифрования

После сборки проектов в папке появляются исполняемые файлы ConsoleApplication1.exe (для шифрования файла) и ConsoleApplication2.exe (соответственно, для дешифрования).

Данные приложения запускаются независимо, так как являются собранными, для запуска приложений не требуется особых прав (при отсутствии политик на ПК, ограничивающих запуск исполняемых файлов на ПК).

Так как в коде программы чётко описан путь для расположения исходных файлов, необходимо наличие каталога C:\Vigenere, который содержит файлы для обработки.

3 Подведение итога разработки приложений шифрования и дешифрования файла

3.1 Практическое применения алгоритмов шифрования и дешифрования

На сегодняшний день, с развитием современных технологий, реализовано очень много различных алгоритмов шифрования и дешифрования. Проще говоря, если мы имеем в представлении схему, которую можно реализовать математически, чтобы потом воплотить в программном коде, то можем разработать собственный алгоритм. Разработка может вестись в любом редакторе кода, который принимает написанную программу как корректную. Реализация данных программ также доступна на других языках семейства C, Java, Python, JavaScript.

На пространстве глобальной сети есть онлайн-эмуляторы различных шифров, которые воспринимают введённый текст, ключ при его необходимости, и выдают результат, но не раскрывают сам механизм шифрования и дешифрования информации.

Подразумевая персональный компьютер как «бездумный исполнитель», возникает необходимость в написании такого программного кода, который не вызовет зацикливания при выполнении в среде на этапе отладки и сборки. В противном случае, при возникновении ошибки, среда разработки откажется выполнять код, указывая на ошибки написания.

Следует воспринимать шифр Виженера как код, представляющий академический интерес, пример построения программы как тренировку при работе с циклами обработки переменных, формирование потоков при считывании информации из файлов и записи в файлы.

В 21 веке шифр Виженера не является устойчивым к хранению данных, передаваемым путём шифрования указанным способом, программными средствами современных операционных систем, программного обеспечения реализованы многие другие механизмы шифрования.

Тем не менее, разработанные программы шифрования и дешифрования позволяют понять процесс шифрования, а также в процессе создания закрепить навыки программирования на языке C++.

Стоит учитывать, что более сложный механизм шифрования/дешифрования требует подключение дополнительных функций, что ведёт к более объёмному коду, написание на языке более высокого уровня предполагает знания для реализации алгоритма на другом языке программирования.

C++ является одним из основных языков программирования современного информационного мира, реализация в формате консольного приложения является

результатом изучения основ программирования на данном языке ранее, которые были расширены в ходе изучения дисциплин по прикладному программированию. Так как понимание логики работы консольных приложений на языке C++ ведёт к расширению практического кругозора, написание программы для шифрования и дешифрования помогает разобраться в работе циклов, заменяющих символы в исходном и конечном файлах.

Профессия C++-разработчика является очень востребованной на сегодняшний день, и написанием консольных приложений не ограничивается применение языка. Реализация приложений на C++ носит скорее служебный характер, действия, которые выполняются в системе и не видные программисту в процессе выполнения.

По информации сайтов с предложениями вакансий C++-разработчиков имеется чёткое разграничение между предложениями от региональных компаний и вакансиями, которые предлагают компании федерального уровня. Сегодня нет чёткой привязки специалистов к своему рабочему месту внутри компании – многие компании предлагают режим удалённой работы. Тем не менее, разработка удалённо возможна при должной квалификации специалиста, при наличии условий труда и возможности доступа к служебным ресурсам. Современные технологии удалённого доступа вполне удовлетворяют требованиям информационной безопасности на предприятиях.

Уровень оплаты труда разработчиков по уровню разработки приведен в таблице 1.

Таблица 1. Уровень оплаты труда программистов C++ по регионам

Локация	Вакансия	Количество имеющихся вакансий	Уровень оплаты труда
Москва	Разработчик уровня Junior	50	60 000
Москва	Разработчик уровня Middle	68	140 000
Москва	Разработчик уровня Senior	567	180 000
Санкт - Петербург	Разработчик уровня Junior	44	60 000
Санкт - Петербург	Разработчик уровня Middle	39	160 000
Санкт - Петербург	Разработчик уровня Senior	295	200 000
Екатеринбург	Разработчик уровня Junior	6	45 000
Екатеринбург	Разработчик уровня Middle	20	80 000
Екатеринбург	Разработчик уровня Senior	45	110 000

Если сравнивать информацию о вакансиях разработчиков на других языках программирования высокого уровня (Java, Python, JavaScript и др.) на тех же сайтах с предложениями о вакансиях, то можно заметить, что цифры немного отличаются не в пользу C++. Это связано с более богатым ассортиментом возможностей реализации запросов рынка, ориентированных на рядового пользователя, использование интерактивного контента для реализации на этих языках.

В связи с развитием операционных систем и программного обеспечения отечественного производства в рамках импортозамещения развитие такого языка как C++ рассматривается в рамках национальных проектов по науке и образованию. Стремительное развитие экономики Российской Федерации в векторе информационных технологий для отечественного рынка не ограничивается созданием ОС Astra Linux для рабочих станций и ОС Аврора для мобильных устройств.

ЗАКЛЮЧЕНИЕ

В процессе написания курсовой работы были изучены материалы по работе консольных приложений на C++, общий принцип реализации алгоритмов шифрования и дешифрования информации.

Проведено изучение источников по криптографии, избранные алгоритмы шифрования приведены в современной интерпретации.

Исследована работа с файлами, содержащими информацию для работы программ.

Реализована работа массивов, содержащих символы алфавита ключа, передача потока информации из программы в файл, расположенный на локальном диске.

Разработаны две программы, позволяющие выполнять операции шифрования/дешифрования.

Перед созданием программы изучены материалы по интересным с образовательной точки зрения шифрам, которые использовались в истории человечества, приведена математическая и математически-логическая схемы шифров.

Разработана блок-схема реализации механизма шифрования/дешифрования информации на языке C++, написаны программы для создания приложений для windows-систем на архитектуре x64.

После написания кода в редакторе MS Visual Studio 2022, произведена отладка кода и сборка проекта, собраны два полноценных *.exe файлы, которые запускаются в системе инициативой пользователя. При запуске приложений в системе не возникают ошибки – окно командной строки корректно отображает текст в необходимой кодировке.

Результаты использования программ приведены в основной части работы, в приложении приведены листинги программ с комментариями разработчиков.

Приведена информация об актуальных предложениях на рынке труда в сфере ИТ Российской федерации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) . Асимметричное шифрование на практике [Электронный ресурс]. – Режим доступа : https://habr.com/ru/post/449552/ (дата обращения 10.03.2022)
2) Бабаш, А. В. История криптографии. Часть I / А.В. Бабаш, Г.П. Шанкин. - М.: Гелиос АРВ, 2019. - 240 с.
3) Баженова, И.Ю. Языки программирования: Учебник для студентов учреждений высш. проф. образования / И.Ю. Баженова; Под ред. В.А. Сухомлин. — М.: ИЦ Академия, 2018. — 368 с.
4) Гавриков, М.М. Теоретические основы разработки и реализации языков программирования: Учебное пособие / М.М. Гавриков, А.Н. Иванченко, Д.В. Гринченков. — М.: КноРус, 2019. — 184 с.
5) Дэвис, С. C++ для «чайников» / С. Дэвис. – М : Диалектика, 2019.
6) Климова, Л. И. C++. Практическое программирование / Л. И. Климова. – М. : Кудиц-Образ, 2001.
7) Кристиансен Т. Perl. Сборник рецептов для профессионалов / Кристиансен Т. – СПб.: Питер, 2004. – 928 с.
8) Назаров, С.В. Современные операционные системы : учебное пособие / С.В. Назаров, А.И. Широков. - М. : Интернет-Университет Информационных Технологий, 2021. - 280 с. : ил., табл., схем. - (Основы информационных технологий).
9) Петров Ю.А. Программирование на языках высокого уровня. Ч. 2: Учебное пособие - Комсомольск-на-Амуре: Комсомольский-на-Амуре гос. техн. ун-т, 2002 – 161 с.
10) Программирование на C и C++ / А. В. Крячков [и др.]. –М. : Горячая линия – Телеком, 2020.
11) Хоффман, Л. Дж. Современные методы защиты информации / Л.Дж. Хоффман. - Москва: СПб. [и др.] : Питер**,** 2018**.** - 264 с.
12) Шилдт Герберт, C++. Руководство для начинающих / Шилдт Герберт. - М.: Диалектика / Вильямс, 2021. - 899 с.
13) Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. - М.: Триумф, 2018. - 518 с.
14) Шифрование поточными методами [Электронный ресурс] - Режим доступа: http://www.netcode.ru/cpp/?artID=4010 (дата обращения 15.03.2022)

15) Шифрование файла C++ [Электронный ресурс] - Режим доступа: https://www.programmersforum.ru/showthread.php?t=293756 (дата обращения 10.03.2022)
16) Алгоритм шифрования DES (версия C ++) [Электронный ресурс] - Режим доступа: https://russianblogs.com/article/49591036147/ (дата обращения 12.03.2022)
17) Шифрование методом Виженера на C++ со сдвигом [Электронный ресурс] - Режим доступа: https://ru.stackoverflow.com/questions/1268827/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BC-%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0-%D1%81%D0%BE-%D1%81%D0%B4%D0%B2%D0%B8%D0%B3%D0%BE%D0%BC-%D0%A1 (дата обращения 16.03.2022)

Приложение А

(справочное)

Текст программ

Текст программы шифрования файла

```
#include <iostream>
#include <conio.h>
#include <algorithm>
#include <fstream>
#include <string>

using namespace std;

void main()
{
     setlocale(LC_ALL,"Russian");
     string Key, RepeatKey, Text, Cipher; // Ключ, Ключ с
повторением, исходный текст

     //Чтение текстового файла и запись его содержимого в строку
для исходного текста
     ifstream F,P;
     F.open("C:\\Vigenere\\Input.txt",ios::in);
      if (F.is_open())
        while (!F.eof())
          getline (F,Text);
        F.close();

        cout <<"Текст из файла: "<<endl<<Text<<endl;

     //Чтение текстового файла и запись его содержимого в строку
для ключа
     P.open("C:\\Vigenere\\Key.txt",ios::in);
     P >> Key;
     P.close();
     cout <<"Ключ из файла: "<<endl<<Key<<endl;

     //Изменение регистра исходного текста и ключа (способ R=1)
     transform(Text.begin(),      Text.end(),      Text.begin(),
::toupper);
     transform(Key.begin(), Key.end(), Key.begin(), ::tolower);
     cout   <<"Измененный   регистр   текста   и   ключа   (способ
R=1):"<<endl;
     cout <<"Исходный текст:"<<endl<<Text<<endl;
     cout <<"Ключ:"          <<endl<<Key <<endl;

     //Перевернем ключ задом на перед
     //в соответствии со способом (P=2)
     reverse(Key.begin(),Key.end());
     cout <<"Ключ задом на перед (способ P=2): "<<endl<<Key<<endl;

     //Ключ с повторением
     for(size_t i=0; i<Text.size(); i++)
     {
          RepeatKey += Key[i % Key.size()];
     }
```

```
     cout<<"Ключ с повторением: "<<endl<<RepeatKey<<endl;

     //Шифрование исходного текста
       //по формуле: C = T + K (mod 26) , где C - зашифрованный
символ, T - символ исходного
     т  екста, K - символ ключа

       for(int i=0; i< Text.size(); i++)
     {
          if(Text[i]==' ')
          {
               Cipher += ' ';
          }
          else
          {
              Cipher  +=  (char)(  (  (  (int)Text[i]  )  +  (
(int)RepeatKey[i] - 32) ) % 26) + 65;
          }
     }
     cout<<"Зашифрованный текст: "<<endl<<Cipher<<endl;

     //Запись зашифрованного текста в файл
     ofstream L;
     L.open("C:\\Vigenere\\Output.txt",ios::out);
     L << Cipher;
     L.close();
     _getch(); }
```

Текст программы дешифрования файла

```
#include <iostream>
#include <conio.h>
#include <algorithm>
#include <fstream>
#include <string>

using namespace std;

void main()
{
     setlocale(LC_ALL,"Russian");
     string Key, RepeatKey, Cipher, DeCipher; // Ключ, Ключ с
повторением, зашифрованный текст, расшифрованный текст

     //Чтение текстового файла и запись его содержимого в строку
для зашифрованного текста
     fstream F,P,L;
     F.open("C:\\Vigenere\\Output.txt",ios::in);
    if (F.is_open())
      while (!F.eof())
          getline (F,Cipher);
    F.close();
     cout <<"Текст из файла: "<<endl<<Cipher<<endl;

     //Чтение текстового файла и запись его содержимого в строку
для ключа
     P.open("C:\\Vigenere\\DeKey.txt",ios::in);
     P >> Key;
     P.close();
```

```
     cout <<"Ключ из файла: "<<endl<<Key<<endl;

     //(Способ R=1)
     transform(Key.begin(), Key.end(), Key.begin(), ::tolower);
     cout    <<"Ключ    переведенный    в    прописные    символы:
"<<endl<<Key<<endl;

     //Перевернем ключ задом наперед
     //в соответствии со способом (P=2)
     reverse(Key.begin(),Key.end());
     cout <<"Ключ задом на перед (способ P=2): "<<endl<<Key<<endl;

     //Ключ с повторением
     for(size_t i=0; i<Cipher.size(); i++)
     {
          RepeatKey += Key[i % Key.size()];
     }
     cout<<"Ключ с повторением: "<<endl<<RepeatKey<<endl;

     //Расшифровка зашифрованного текста
     //по формуле: T = C - K + 26 (mod 26) , где C - зашифрованный
символ,
          T - символ исходного текста, K - символ ключа
     for(int i=0; i< Cipher.size(); i++)
     {
          if(Cipher[i] == ' ')
          {
               DeCipher += ' ';
          }
          else
          {
       DeCipher += (char)( ( (int)Cipher[i] - ( (int)RepeatKey[i]
- 32 ) + 26 ) % 26) + 65;
          }
     }
     cout<<"Зашифрованный текст: "<<endl<<DeCipher<<endl;

     //Запись зашифрованного текста в файл
     L.open("C:\\Vigenere\\Decipher.txt",ios::out);
     L << DeCipher;
     L.close();
     _getch();
}
```

Приложение Б

Скриншот с сайта Антиплагиат.

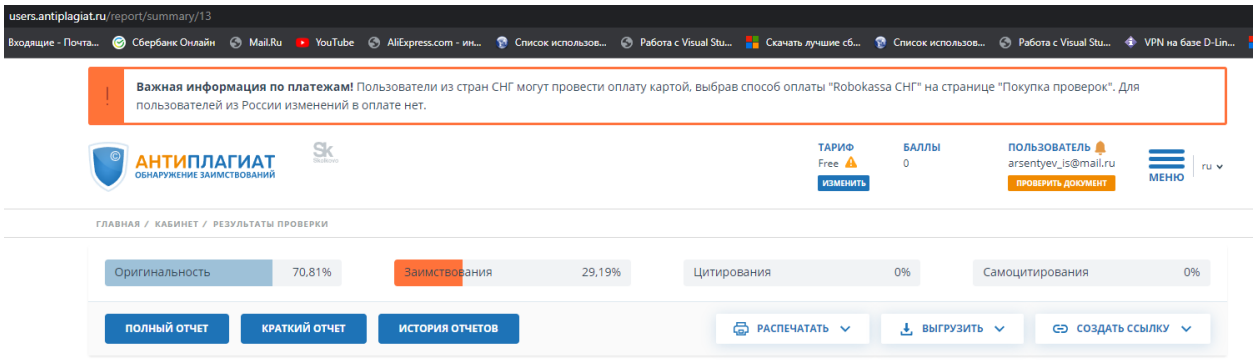


Рисунок 9 – Скрин с сайта Антиплагитат(https://www.antiplagiat.ru)