САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

Тема: Программная реализация книжного шифра

Студент гр. 8307 _____ Никулин Л.А.

Преподаватель _____ Перязева Ю.В.

Санкт-Петербург
2018

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студент Никулин Л.А.

Группа 8307

Тема работы: Программная реализация книжного шифра

Основные требования:

Используя знания, полученные в ходе семестра по курсу программирования, реализовать в программе, написанной на языке программирования Си, шифрование и дешифрование по книжному шифру.

Содержание пояснительной записки:

- Введение
- История и описание книжного шифра
- Программная реализация шифра
- Заключение
- Список используемых источников
- Приложения

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 08.12.2018

Дата сдачи реферата: 21.12.2018

Дата защиты реферата: 24.12.2018

Студент _____ Никулин Л.А.

Преподаватель _____ Перязева Ю.В.

АННОТАЦИЯ

В данной курсовой работе была рассмотрена реализация книжного шифра на языке программирования Си. Для создания программы, была изучена история криптографии и книжного шифра в частности. В результате исследований и разработок программного кода, была получена и протестирована программа, позволяющая кодировать и декодировать текст с помощью какой-либо книги и записывать результаты в текстовые файлы.

SUMMARY

In this course work was considered the implementation of the book cipher in the C programming language. To create the program, the history of cryptography and the book cipher in particular was studied. As a result of research and development of software code, a program was obtained and tested, which allows to encode and decode text using any book and record the results in text files.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Широкое применение компьютерных технологий и постоянное увеличение объема информационных потоков вызывает постоянный рост интереса к криптографии. В курсовой работе рассматривается программная реализация книжного шифра, позволяющего шифровать и расшифровывать информацию. При разработке программного кода и реализации исторических шифров были закреплены знания, полученные в ходе обучения.

Цели работы:

- Разработать программу шифрования и дешифрования при помощи книжного шифра
- Закрепить навыки, полученные во время прохождения курса программирования.

Задачи работы:

- Изучить историю криптографии и книжного шифра
- Изучить принципы шифрования и дешифрования
- Построить блок-схему программы
- Написать программу, реализующую шифрование и дешифрование
- Протестировать программу и убедиться в её корректности.

Книжный шифр

История шифра

Книжный шифр — вид шифра, в котором каждый элемент открытого текста (каждая буква или слово) заменяется на указатель (например, номер страницы, строки и столбца) аналогичного элемента в дополнительном тексте-ключе.

Для дешифрования необходимо иметь как закрытый текст, так и дополнительный текст-ключ. В качестве дополнительного текста часто использовали распространённые книги, либо книги, которые с большой долей вероятности были и у отправителя, и у адресата.

На сегодняшний день ученые и историки не имеют четкой информации о том, когда и кем первый раз был использован книжный шифр. Один из наиболее ранних примеров использования данного шифра известен как книжный шифр Энея, который относится к стеганографии. Возможно, что эта первая попытка использовать рукописный текст для шифрования и стала началом создания книжного шифра.

Кроме того, нельзя не заметить сходство использования книжного шифра и шифрования с помощью квадрата Полибия. Причём книжный шифр — один из методов предложенных Полибием, только значительно усовершенствованный. В квадрате Полибия каждой букве в соответствие ставятся два числа, причём для одинаковых букв числа также будут идентичны. Преимущество книжного шифра в том, что каждая буква исходного текста будет иметь свой собственный идентификатор. Однако, если, например, страницу книги разбить на большое число различных квадратов Полибия, то системы шифрования будут одинаковыми.

Позже над изучением и усовершенствованием книжного шифра работали в 1849 Мейснер в Брауншвейге, а в новейшее время Вольтер в Винтертуре. В Советском Союзе Книжный шифр широко использовался и исследовался партией большевиков РСДРП(б). Создание нескольких его модификаций приписывается известной большевичке Елене Дмитриевне Стасовой.

Описание шифра

Знакомство с книжными шифрами следует начать с простого книжного шифра.

Перед началом работы необходимо выбрать книгу и страницу, которые будут использоваться для шифрования. Это может быть, например, первая страница первой части третьего тома романа «Война и мир» Льва Николаевича Толстого. Для шифрования простого сообщения достаточно воспользоваться первыми двумя абзацами текста.Все слова текста, напечатанного на этой странице бессмертного произведения великого русского классика, за исключением дат, следует пронумеровать. В результате текст примет следующий вид:

«С(1) конца(2) 1811 года(3) началось(4) усиленное(5) вооружение(6) и(7) сосредоточение(8) сил(9) Западной(10) Европы(11), и(12) в(13) 1812году(14)силы(15)эти(16) — миллионы(17) людей(18) (считая(19) тех(20), которые(21) перевозили(22) и(23) кормили(24) армию(25),) двинулись(26) с(27) Запада(28) на(29) Восток(30), к(31) границам(32) России(33), к(34) которым(35) точно(36) так(37) же(38) с(39) 1811 года(40) стягивались(41) силы(42) России(43). 12 июня(44) силы(45) Западной(46) Европы(47) перешли(48) границы(49) России(50), и(51) началась(52) война(53), то(54) есть(55) совершилосъ(56) противное(57) человеческому(58) разуму(59) и(60) всей(61) человеческой(62) природе(63) событие(64). Миллионы(65) людей(66) совершали(67) друг(68) против(69) друга(70) такое(71) бесчисленное(72) количество(73) злодеяний(74), обманов(75), измен(76), воровства(77), подделок(78) и(79) выпуска(80) фальшивых(81) ассигнаций(82), грабежей(83), поджогов(84) и(85) убийств(86), которого(87) в(88) целые(89) века(90) не(91) соберет(92) летопись(93) всех(94) судов(95) мира(96) и(97) на(98) которые(99) в(100) этот(101) период(102) времени(ЮЗ) люди(104), совершавшие(105) их(106), не(107) смотрели(108) как(109) на(110) преступления(111).

Что(112) произвело(113) это(114) необычайное(115) событием(116)? Какие(117) были(118) причины(119) его(120)? Историки(121) с(122) наивнойМ23) уверенностью(124) говорятМ25), что(126) причинамиМ?7) этого(128) события(129) были(130) обида(131), нанесеннаяМ32) герцогу(133) Ольденбургскому(134), несоблюдение(135) континентальной(136) системы(137), властолюбие(138) Наполеона(139), твердость(140) Александра(141), ошибки(142) дипламатов(143) ит.(144) п.(145).

Алгоритм шифрования при использовании простого книжного шифра заключается в том, что цифра 1 обозначает первую букву первого слова, то

есть в рассматриваемом примере букву С. Цифра 2 соответствует первой букве второго слова — букве К и так далее. Например, число 38 соответствует букве Ж, а число 81 — букве Ф.

Наблюдательный читатель заметит, что одной и той же букве соответствуют разные числа. Так, например, букве Г соответствуют числа 3,14,32 и др. В этом заключается одно из достоинств книжного шифра. Поскольку одну и ту же букву открытого текста в криптограмме можно заменить разными числами, разгадать такую криптограмму с помощью методов частотного анализа невозможно.

В качестве примера попробуем зашифровать с помощью простого книжного шифра открытый текст СЕКРЕТНОЕ ПОСЛАНИЕ. Итак, если в данном открытом тексте заменить буквы на соответствующие им числа из приведенного выше текста, то полученная криптограмма будет выглядеть так:

1.11.87.33.47.71.107.75.55. 22.134.108.93.25.91.121.120

Для того чтобы расшифровать это сообщение, получатель должен в аналогичной книге на известной ему странице пронумеровать все слова, а затем произвести замену указанных в криптограмме чисел на соответствующие буквы.

Усовершенствованный книжный шифр

При практическом применении рассмотренного ранее простого книжного шифра пользователь, *без* сомнения, столкнется с одной трудно разрешимой проблемой. Она заключается в том, что в русском алфавите есть буквы, с которых начинается лишь небольшое число слов, таких как, например, буква Ы. Найти такие слова в подавляющем числе книг просто невозможно. В то же время в русском языке практически вообще нет слов, которые начинались бы с таких букв, как Ъ или Ь. Однако незначительное усовершенствование простого книжного шифра позволяет решить эту задачу.

В усовершенствованном книжном шифре для замены каждой буквы открытого текста используются два числа, записываемые через тире. При этом первое число означает порядковый номер слова в тексте, а второе число означает номер буквы в этом слове.

Так, например, в приведенном ранее тексте первой страницы первой части третьего тома романа «Война и мир» Л. Н. Толстого число 2–4 соответствует четвертой букве второго слова, то есть букве Ц Таким же

образом определяются числа для других букв. Число 46-8 соответствует в данном темпе букве Й, число 134-3 — букве Ь, число 49-7 — букве Ы и так далее.

Теперь, если в открытом тексте СЕКРЕТНОЕ ПОСЛАНИЕ заменить буквы на соответствующие им числа в соответствии с рассматриваемым алгоритмом шифрования, то полученная криптограмма будет выглядеть так:

4–7.48-2.117-1.83-2.89-5.137-4.57-7.101-3.67-4. 48-1.123-6.82-3.74-2.117-2.124-7.119-3.20-2

Расшифровка такой криптограммы для получателя сообщения не представляет труда. Достаточно в аналогичной книге на определенной странице пронумеровать все слова, а затем произвести замену указанных в криптограмме чисел на соответствующие буквы. В то же время несанкционированный пользователь разгадать подобную шифрограмму не сможет.

Программная реализация

Описание решения:

Нам даны 2 файла, находящие в корневом разделе нашего проекта: text.txt и book.txt. В первом находиться текст, который мы хотим зашифровать, во втором – книга, по которой мы будем шифровать. Вся программа состоит из трёх функций – encryption, decryption и main.

Функция encryption реализует шифрование исходного текста и выводит результат в cipher.txt. Сначала открывается 2 файла на чтение (text.txt, book.txt) и один файл на запись (cipher.txt). При ошибке открытия какого-либо файла выводиться уведомление о неудаче открытия. Далее, при помощи двух вложенных циклов, реализуется поиск каждого символа из исходного текста в книге. Пока не нашёлся нужный нам символ, указатель двигается по файлу, сравнивая каждый символ с искомым. Параллельно с этим считается количество пробелов, которые мы встретили на своём пути. Если был встречен пробел, увеличиваем счётчик nomerslova на 1 (изначально он равен единице), а также запоминаем позицию последнего найденного пробела в переменной pos_last_space. Когда мы нашли нужный нам символ, запоминаем его позицию. Далее, записываем через пробел в файл cipher.txt значение nomerslova и pos_symblol - pos_last_space. Таким образом первое записанное число – это номер слова, в каждом найдена нужная нам буква, а второе – номер буквы в этом слове (так как разность позиций символа во всем тексте и последнего пробела до этого слова как раз даёт нам номер буквы в слове). Числа каждого последующего числа записываются на новой строке. Повторяем все эти операции пока не закодируем каждый символ исходного текста. Далее выводим сообщение об успешном шифровании и закрываем все открытые ранее файлы. При ошибке закрытия какого-либо файла выводиться уведомление о неудаче закрытия.

Функция decryption реализует дешифрование шифра и выводит результат в newtext.txt. Сначала открывается 2 файла на чтение (cipher.txt, book.txt) и один файл на запись (newtext.txt). При ошибке открытия какого-либо файла выводиться уведомление о неудаче открытия. Далее, при помощи двух вложенных циклов, реализуется поиск каждого символа, закодированного двумя цифрами в книге. В каждой строке файле cipher.txt записаны два числа. Первое число (last_space) отвечает за номер слова. Поэтому, мы передвигаем указатель в файле-книге на 1 символ вправо до тех пор, пока не встретился пробел. При нахождения пробела, увеличиваем счётчик пробелов n_space на 1. Повторяем это до тех пор, пока n_space не станет равным last_space-1. Второе число (pos_symbol) отвечает за номер символа в слове. Поэтому, мы передвигаем указатель в книге на значение этой переменной от того места, где

был найден последний пробел при помощи функции fseek(). Считываем символ, записанный на этом месте, а потом записываем его в newtext.txt. Далее считываем 2 числа на следующей строке файла cipher.txt. Повторяем эти операции, пока не будет расшифрованы все буквы. Далее выводим сообщение об успешном дешифровании и закрываем все открытые ранее файлы. При ошибке закрытия какого-либо файла выводиться уведомление о неудаче закрытия.

В функции main реализован диалог с пользователем и вызов вышеописанных функций. Сначала выводиться инструкция по работе с программой. Далее пользователь выбирает одно из действий, нажимая на соответственную цифру. При нажатии на ‘1’, шифруется исходные текст при помощи функции encryption. При нажатии на ‘2’, дешифруется шифр при помощи функции decryption. При нажатии на ‘3’, очищается консоль. Если был введён какой-либо другой знак, программа уведомит об этом и попросит повторить ввод. При нажатии на ‘0’ программа прекращает свою работу и уведомляет об этом.

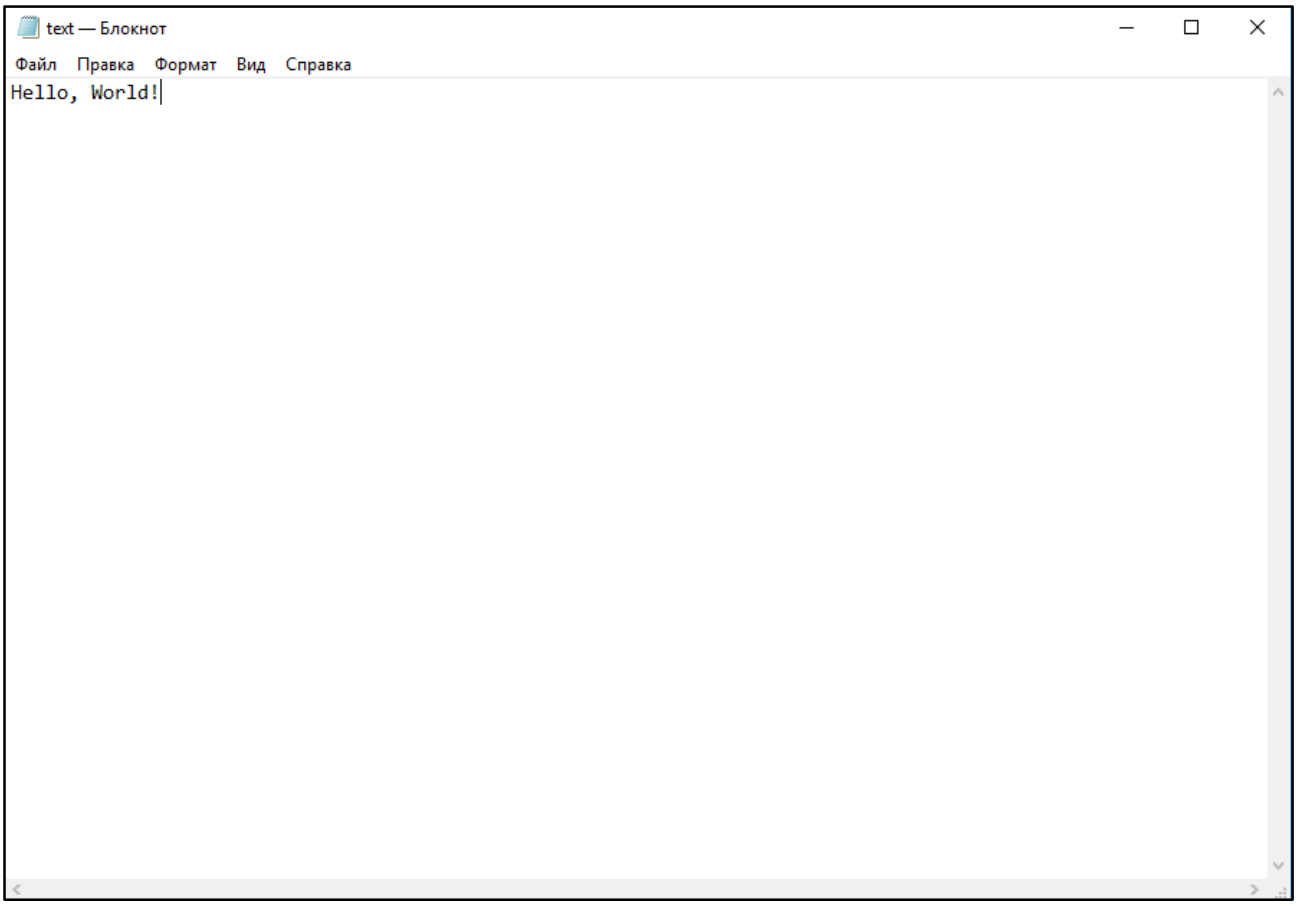
Описание переменных:

Имя переменной	Тип	Назначение
int main ()		
operation	int	Принимает режим работы программы, введенного пользователем
void encryption()		
pO, pB, pC	FILE *	Указатели на файлы исходного текста, книги и шифра соответственно
pos_last_space	long	Запоминает позицию последнего пробела
pos_symblol	long	Запоминает позицию искомого символа
cO, cB	char	Считывание 1 символа из исходного текста, книги соответственно
nomerslova	int	Запоминает номер слова
flag	int	Флаг для выхода из цикла (если нужный символ найден, переходим к следующему)
void decryption()		
pB, pC, pN	FILE *	Указатели на файлы книги, шифра и раскодированного текста соответственно
last_space	int	Количество пробелов до нужного слова
n_space	int	Счётчик пробелов
pos_symbol	long	Позиция символа в слове
cc	char	Считывание и запись 1 символа из книги в newtext.txt

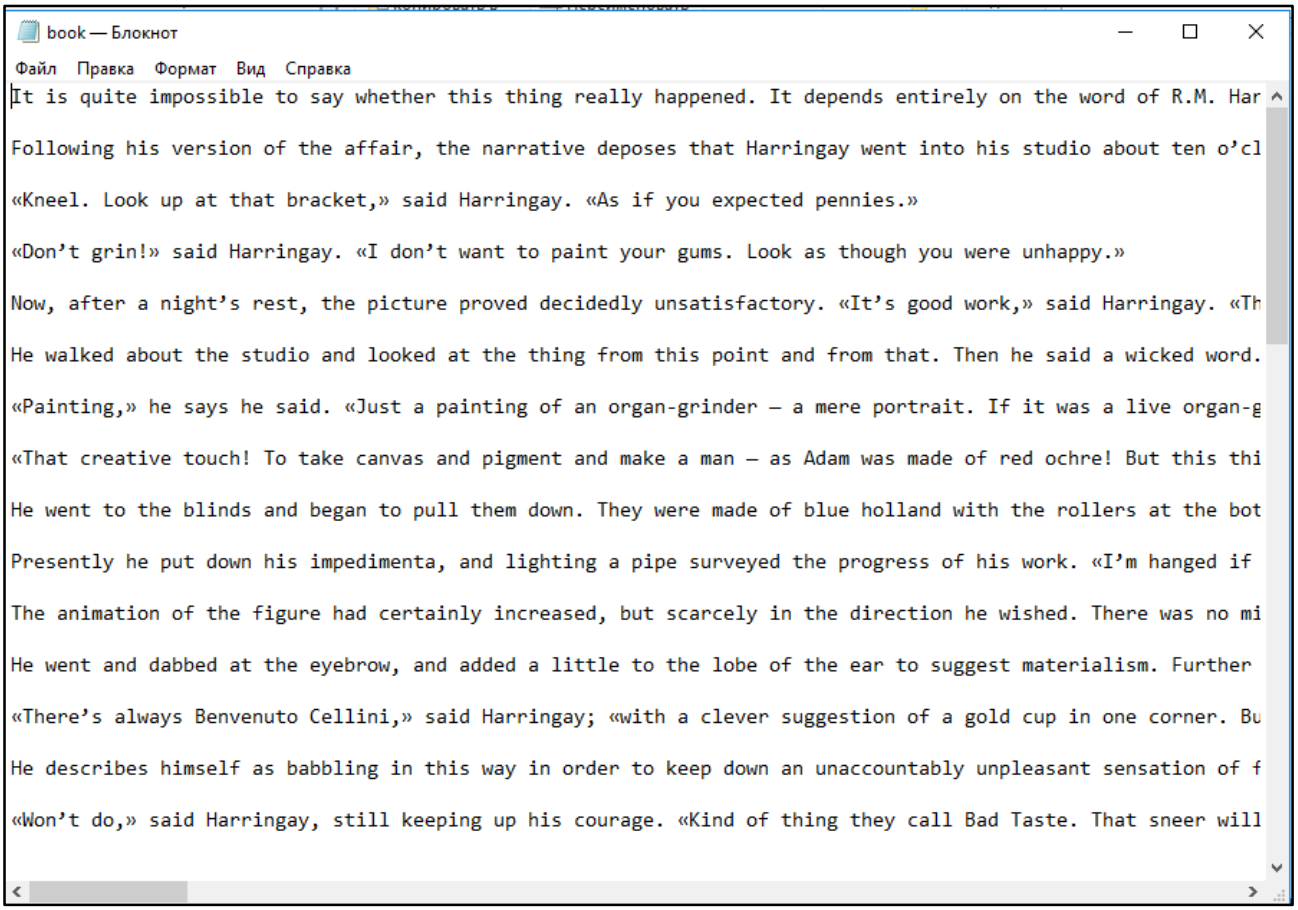
Контрольные примеры:

№1.

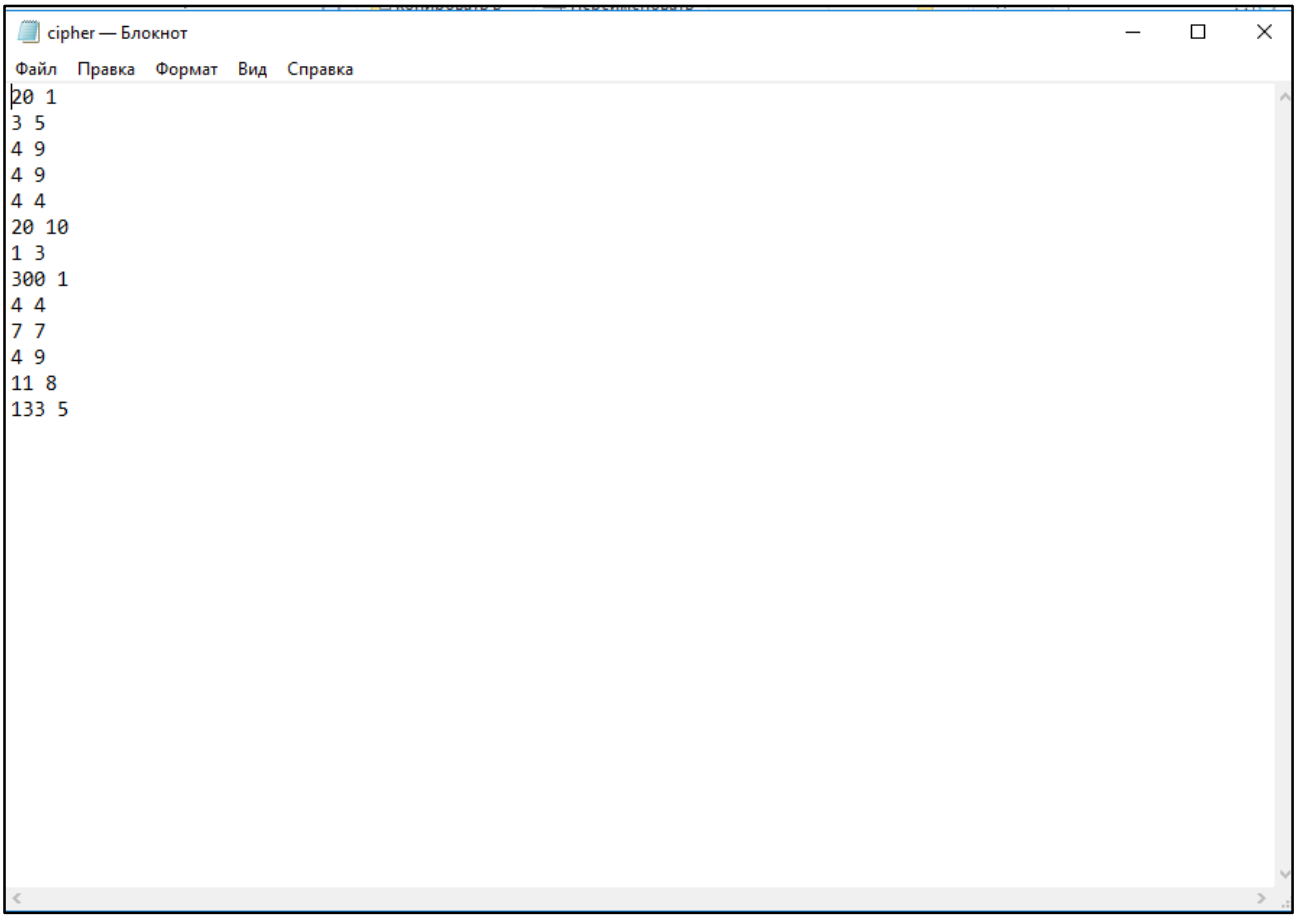
Шифрование:



Исходный текст (1).

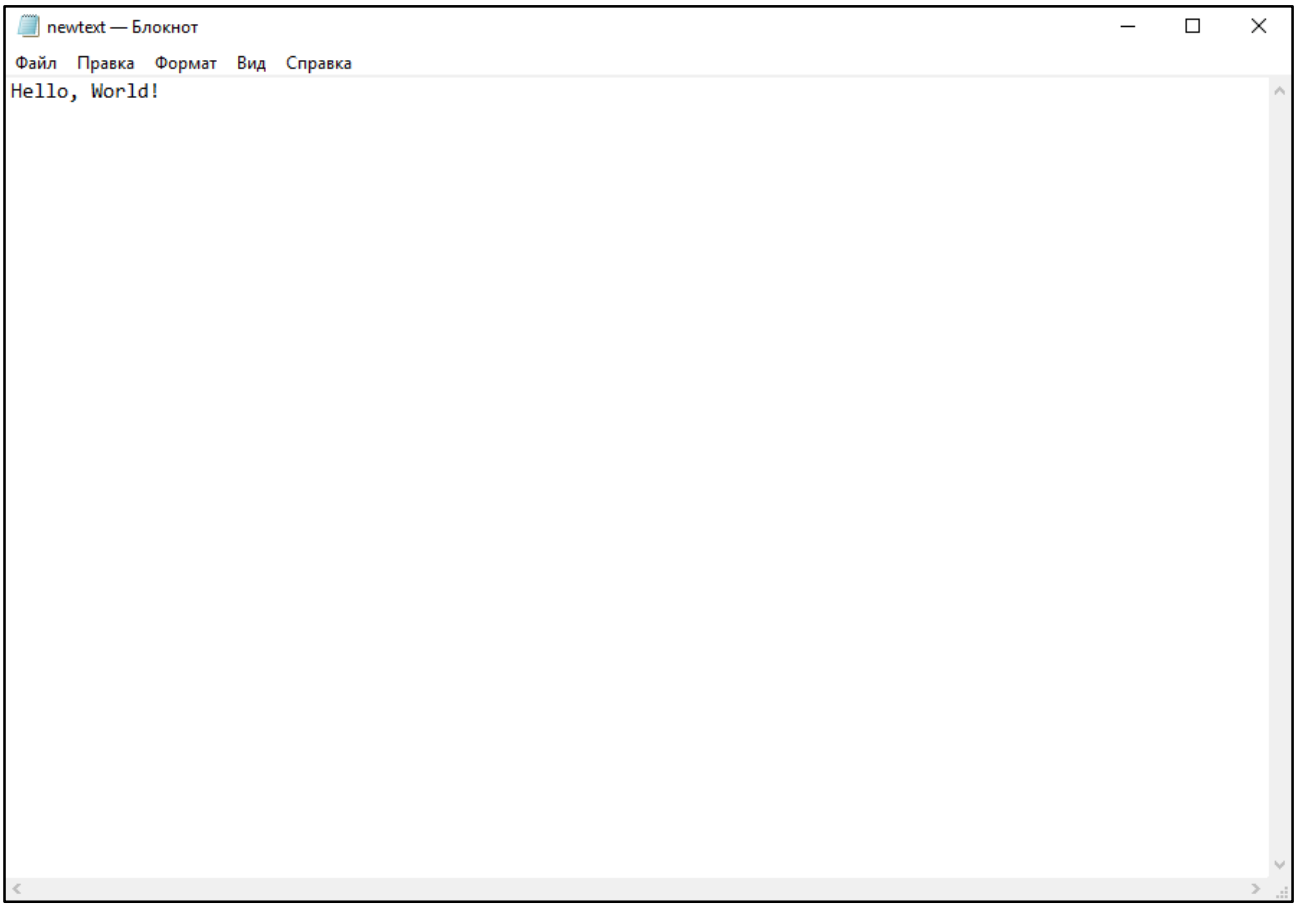


Книга (1).



Полученный шифр (1).

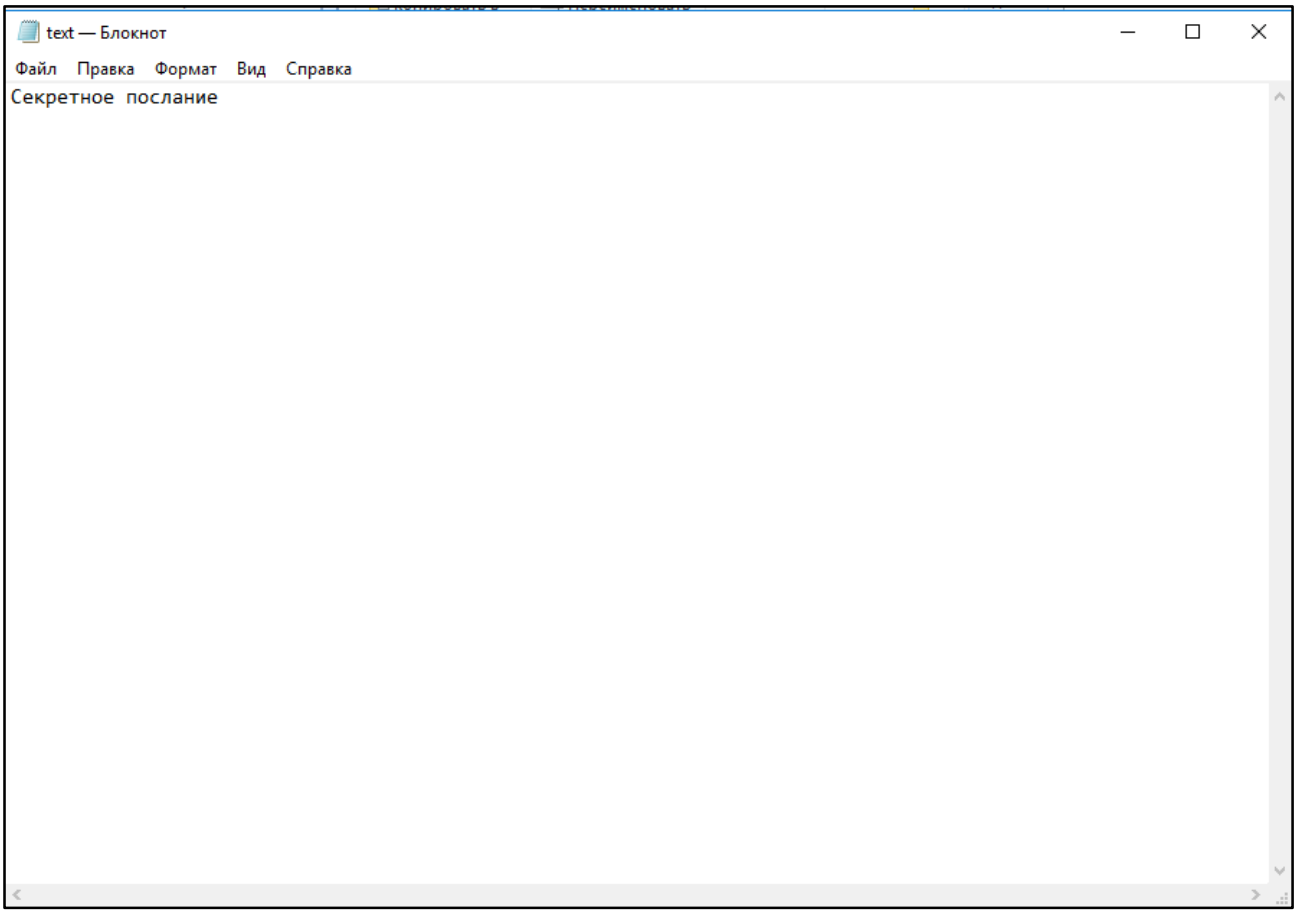
Дешифрование:



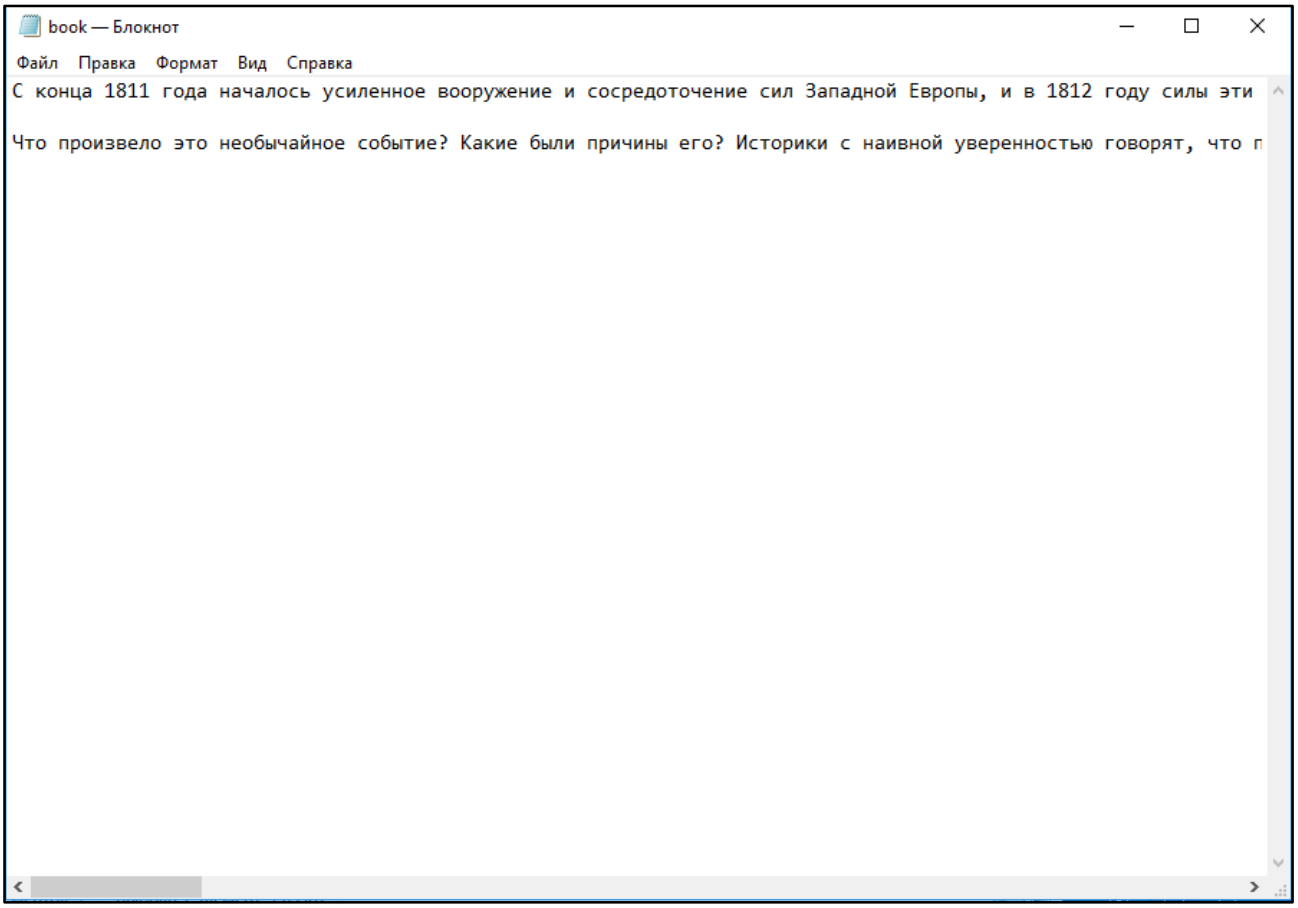
Расшифрованное сообщение (1).

№2.

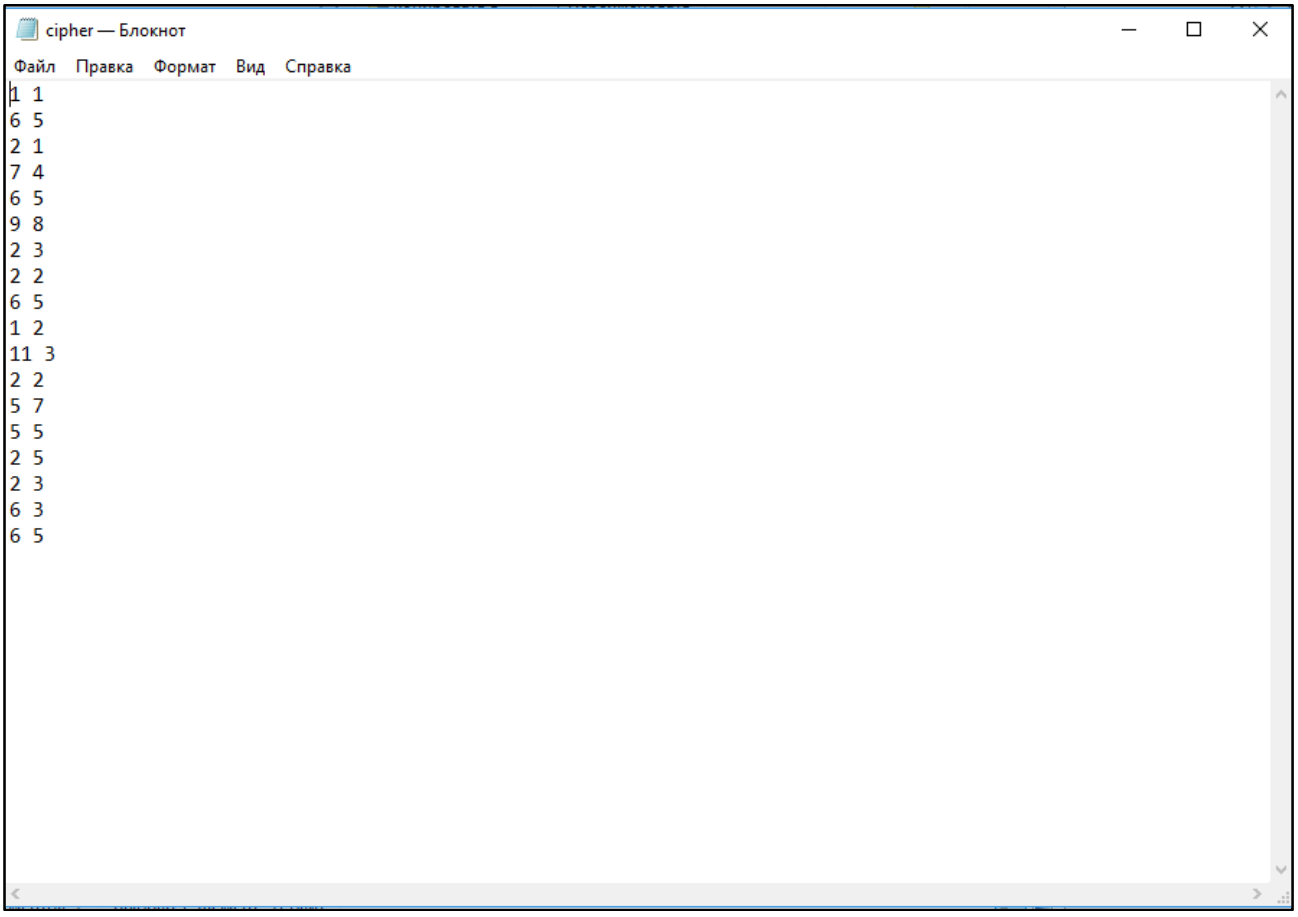
Шифрование:



Исходный текст (2).

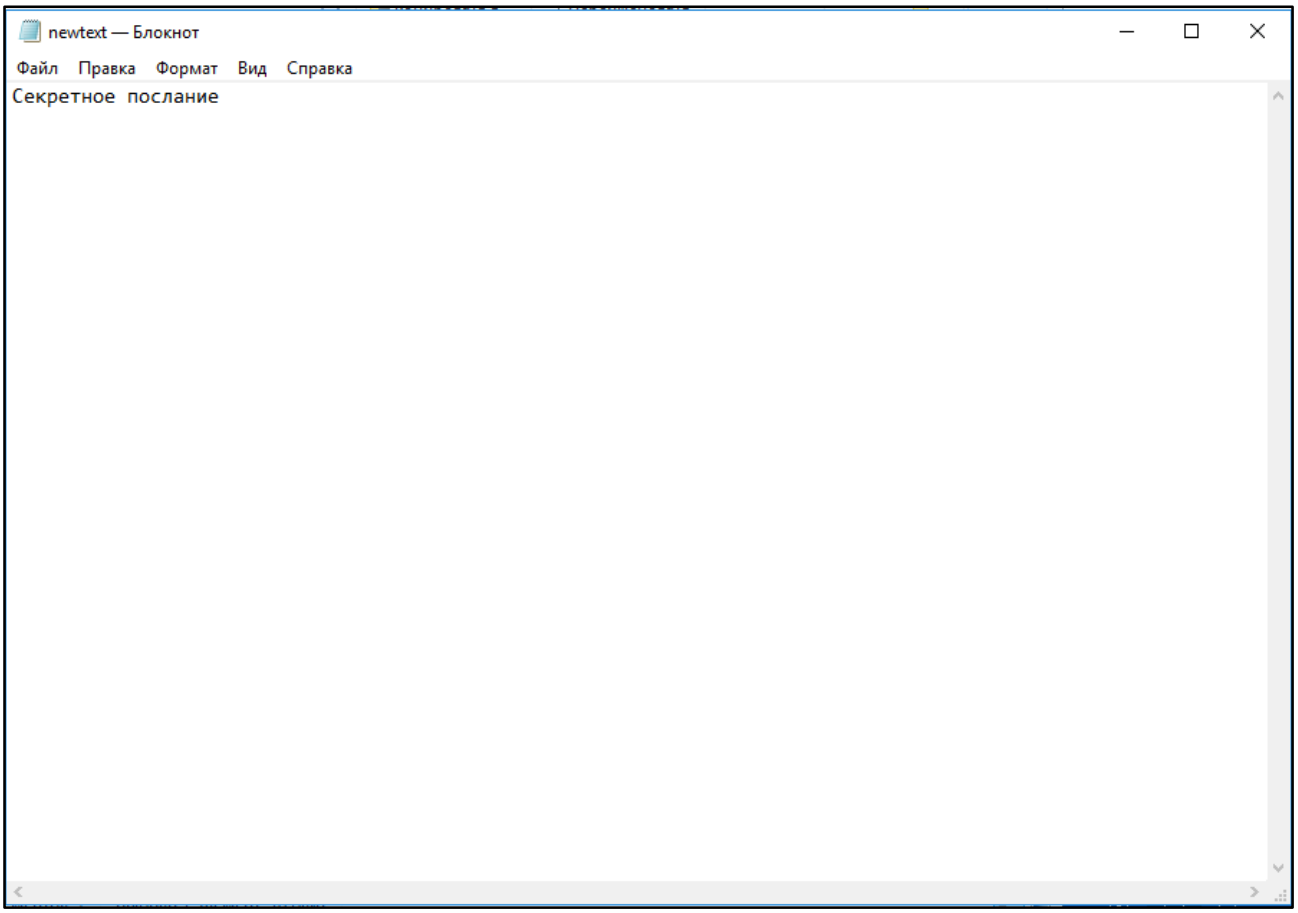


Книга (2).



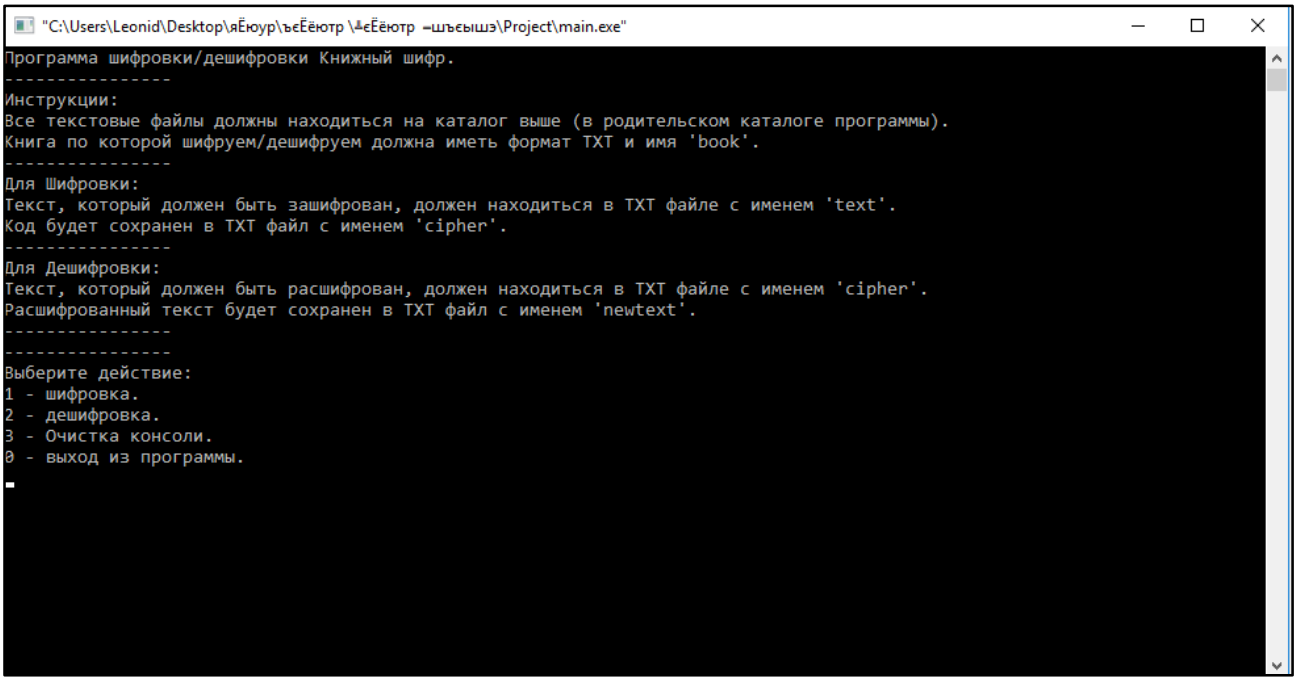
Полученный шифр (2).

Дешифрование:

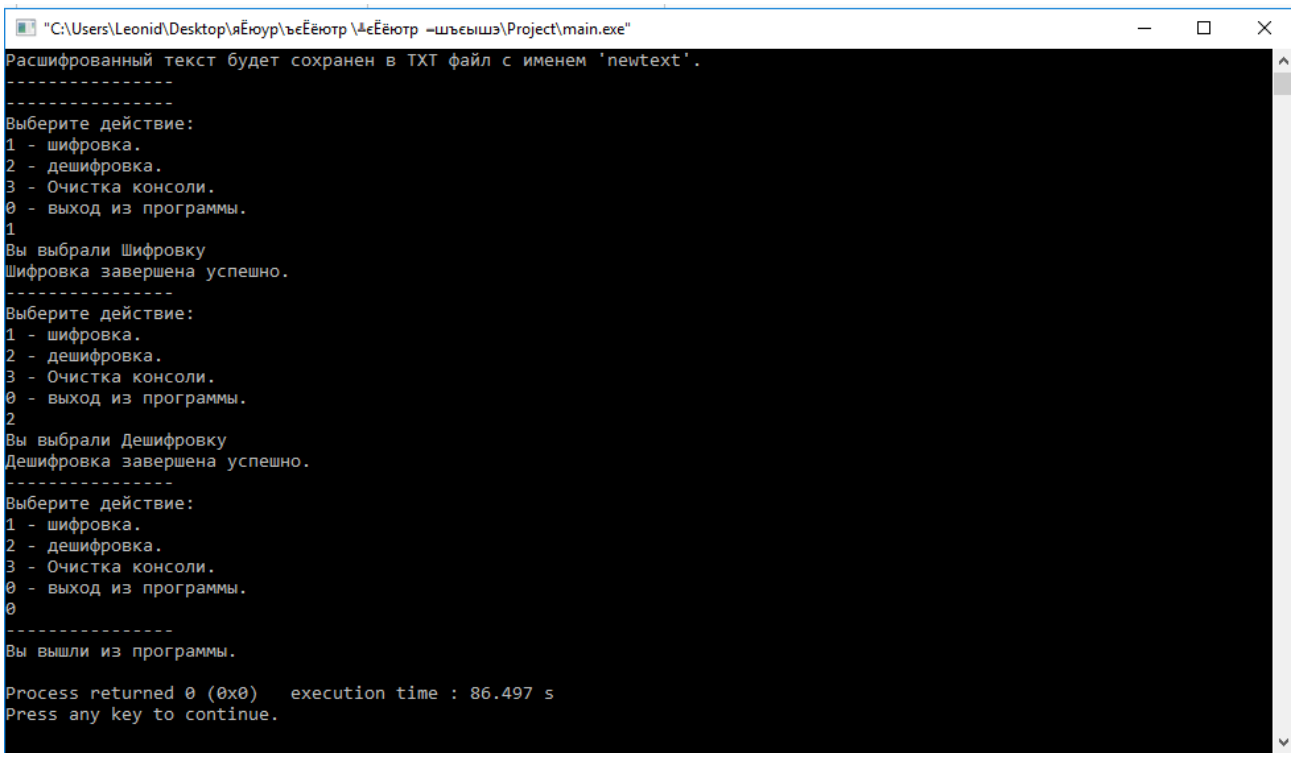


Расшифрованное сообщение (2).

Примеры работы программы:



Пример 1.



Пример 2.

ЗАКЛЮЧЕНИЕ

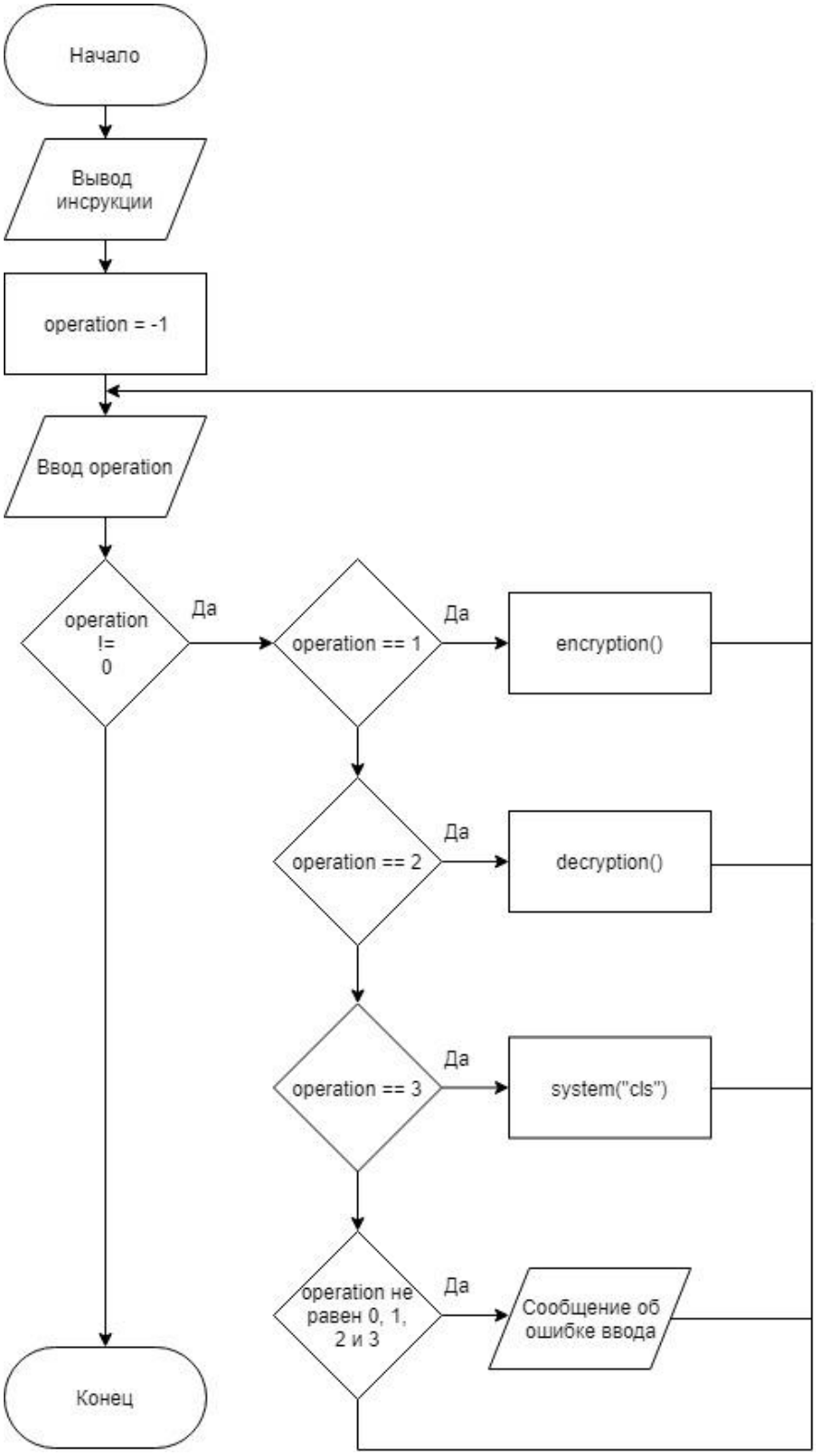
Были достигнуты все намеченные цели – была создана программа на языке программирования Си для шифрования и дешифрования текста при помощи книжного шифра. Также, были закреплены на практике все знания, которые были получены в течении семестра. Приложение было неоднократно протестировано на разных книгах и исходных текстах, которые имели разные размеры и языки написания (английский и русский). Все поставленные задачи были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

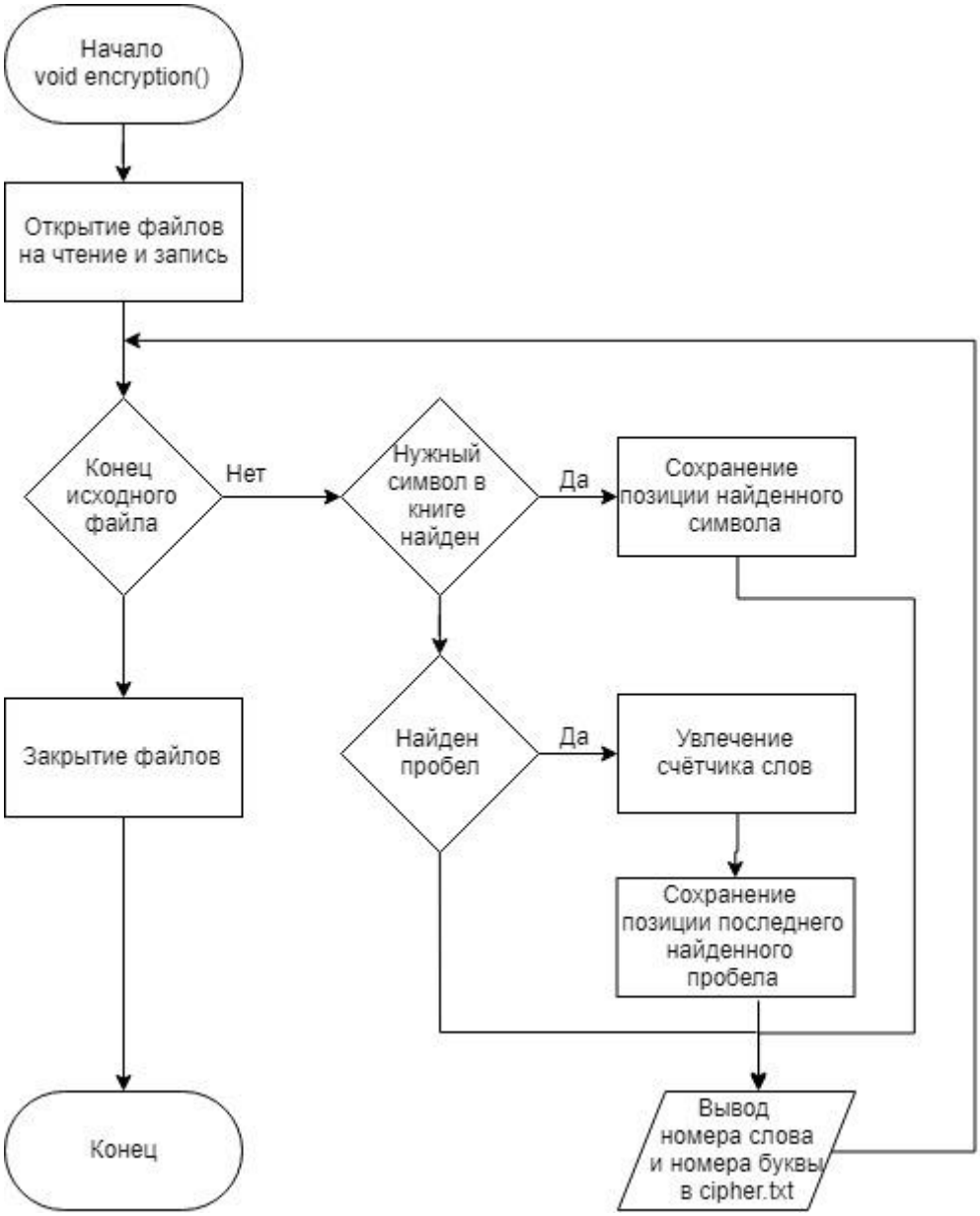
1. *Керниган Б. В., Ритчи Д. М.* Язык программирования Си: Пер. с англ. — 3-е изд. — СПб.: Невский Диалект, 2001. — 352 с.
2. wm-help.net/lib/b/book/3167518757/165 – Книжный шифр (дата обращения 09.12.2018)
3. http://cert.obninsk.ru/gost/282/282.html - ГОСТ 19.701-90. СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ ДАННЫХ И СИСТЕМ (дата обращения 10.12.2018).

ПРИЛОЖЕНИЕ А

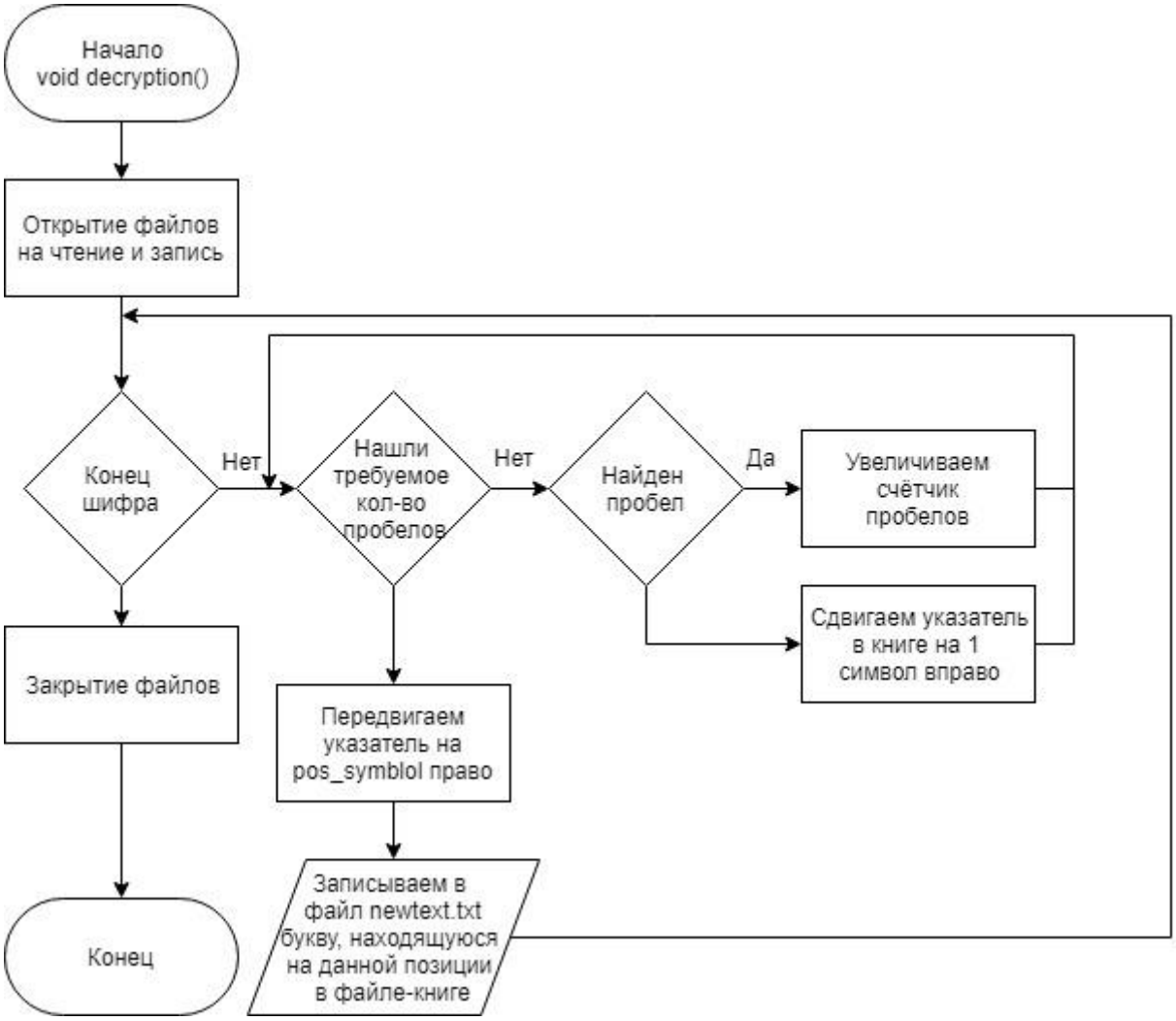
БЛОК-СХЕМА



int main()



void encryption()



void decryption()

ПРИЛОЖЕНИЕ Б

ТЕКСТ ПРОГРАММЫ

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <locale.h>

void decryption()
{
    FILE *pN, *pB, *pC;
    int last_space, n_space = 0;
    long pos_symbol;
    char cc;

    if ((pN = fopen("../newtext.txt", "w")) == NULL)
    {
        printf("Не удалось открыть файл для записи, повторите
попытку");
    }
    if ((pB = fopen("../book.txt", "r")) == NULL)
    {
        printf("Не удалось открыть книгу, повторите попытку");
    }
    if ((pC = fopen("../cipher.txt", "r")) == NULL)
    {
        printf("Не удалось открыть оригинальный текст, повторите
попытку");
    }

    while (fscanf(pC, "%d%ld", &last_space, &pos_symbol) != EOF)
    {
        fseek(pB, 0, SEEK_SET);
        while (n_space < last_space-1)
        {
            fscanf(pB, "%c", &cc);
            if (cc == ' ') n_space++;
        }
        fseek(pB, pos_symbol-1, SEEK_CUR);
        fscanf(pB, "%c", &cc);
        fprintf(pN, "%c", cc);
        n_space = 0;
     }
    puts("Дешифровка завершена успешно.");

    if(fclose(pN)==EOF) perror("Input closing error");
    if(fclose(pB)==EOF) perror("Input closing error");
    if(fclose(pC)==EOF) perror("Input closing error");
}

void encryption()
{
    FILE *pO, *pB, *pC;
    long pos_last_space = 0, pos_symblol;
    char cO, cB;
    int nomerslova = 1;
```

```
    int flag = 0;

    if ((pO = fopen("../text.txt", "r")) == NULL)
    {
        printf("Не удалось открыть оригинальный текст, повторите
попытку");
    }
    if ((pB = fopen("../book.txt", "r")) == NULL)
    {
        printf("Не удалось открыть книгу, повторите попытку");
    }
    if ((pC = fopen("../cipher.txt", "w")) == NULL)
    {
        printf("Не удалось открыть файл для записи, повторите
попытку");
    }

    while(((cO=fgetc(pO))!=EOF))
    {
        fseek(pB, 0, SEEK_SET);
        while(((cB=fgetc(pB))!=EOF) && (flag ==0))
        {
           if(cO==cB)
           {
               pos_symblol = ftell(pB);
               fprintf(pC,"%d %ld\n", nomerslova, pos_symblol -
pos_last_space);
               flag = 1;
           }
           if(cB==' ')
           {
               nomerslova+=1;
               pos_last_space = ftell(pB);
           }
        }
        flag = 0;
        nomerslova = 1;
        pos_last_space = 0;
    }
    puts("Шифровка завершена успешно.");

    if(fclose(pO)==EOF) perror("Input closing error");
    if(fclose(pB)==EOF) perror("Input closing error");
    if(fclose(pC)==EOF) perror("Input closing error");
}

int main()
{
    setlocale(LC_ALL, "RUS");
    int operation = -1;
    puts("Программа шифровки/дешифровки Книжный шифр.");
    puts("----------------");
    puts("Инструкции:");
    puts("Все текстовые файлы должны находиться на каталог выше
(в родительском каталоге программы).");
    puts("Книга по которой шифруем/дешифруем должна иметь формат
TXT и имя 'book'.");
     puts("----------------");
    puts("Для шифровки:");
```

```
    puts("Текст, который должен быть зашифрован, должен
находиться в TXT файле с именем 'text'.");
    puts("Код будет сохранен в TXT файл с именем 'cipher'.");
    puts("----------------");
    puts("Для Дешифровки:");
    puts("Текст, который должен быть расшифрован, должен
находиться в TXT файле с именем 'cipher'.");
    puts("Расшифрованный текст будет сохранен в TXT файл с
именем 'newtext'.");
    puts("----------------");
    puts("----------------");
    while(operation != 0)
    {
        puts("Выберите действие:");
        puts("1 - шифровка.");
        puts("2 - дешифровка.");
        puts("3 - Очистка консоли.");
        puts("0 - выход из программы.");
        if (scanf("%d", &operation) != 1) // если данные не
удалось присвоить переменной,
            scanf("%*s"); // то выбросить их в виде строки.
        if (operation != 1 && operation != 2 && operation != 3
&& operation != 0)
        {
          puts("Выбранного действия не существует, проверьте
правильность ввода и повторите попытку.");
          operation = -1;
          }
        else if (operation == 1)
        {
            printf("Вы выбрали Шифровку\n");
            encryption();
        }
        else if (operation == 2)
        {
            printf("Вы выбрали Дешифровку\n");
            decryption();
        }
        else if (operation == 3)
        {
            system("cls");
        }
        puts("----------------");
    }

    puts("Вы вышли из программы.");
    return 0;
}
```