Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Основы программирования и алгоритмизации

УЧЕБНАЯ ПРАКТИКА

Студент гр. 150503 И. И. Федорович

Руководитель П. А. Дулько

МИНСК 2022

СОДЕРЖАНИЕ

1.Введение……3
2.Криптография……...4
3.Частотный анализ……6
4.Шифрование……..7
5.Шифр Цезаря……8
6.Листинг кода……...10
7.Блок-схемы некоторых функций……...……...17
Заключение……20

1.Введение

Данная работа написана с целью освящения темы “Частотный анализ”. В работе использован классический метод реализации частотного анализа, а также выполнено шифрование и дешифрование англоязычного текст. По итогу шифрование получен частотный анализ текста и выявлена зависимость шифрования. В работе использован метод шифрование Цезаря. В работе представлена теоретическая информация о частотном анализе, шифровании методом Цезаря, листинг кода, реализованный на языке Си, блок-схемы некоторых функций, представленных в программе.

2.Криптография

Криптогра́фия (от др.-греч. κρυπτός «скрытый» + γράφω «пишу») — наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), шифрования (кодировка данных).

Изначально криптография изучала методы шифрования информации — обратимого преобразования открытого (исходного) текста на основе секретного алгоритма или ключа в шифрованный текст (шифротекст).

Традиционная криптография образует раздел симметричных криптосистем, в которых зашифровывание и расшифровывание проводится с использованием одного и того же секретного ключа.

Для современной криптографии характерно использование открытых алгоритмов шифрования, предполагающих использование вычислительных средств. Известно более десятка проверенных алгоритмов шифрования, которые при использовании ключа достаточной длины и корректной реализации алгоритма криптографически стойки. Распространённые алгоритмы:

- симметричные DES, AES, Camellia, Twofish, Blowfish, IDEA, RC4 и др.;
- асимметричные RSA и Elgamal (Эль-Гамаль);
- хеш-функций MD4, MD5, MD6, SHA-1, SHA-2, ГОСТ Р 34.11-2012 («Стрибог») .

Криптографические методы стали широко использоваться частными лицами в электронных коммерческих операциях, телекоммуникациях и многих других средах.

В основе построения криптостойких систем лежит многократное использование относительно простых преобразований, так называемых криптографических примитивов. Клод Шеннон известный американский математик и электротехник предложил использовать подстановки (англ. substitution) и перестановки (англ. permutation). Схемы, которые реализуют эти преобразования, называются SP-сетями. Нередко используемыми криптографическими примитивами являются также преобразования типа циклический сдвиг или гаммирование. Ниже приведены основные криптографические примитивы и их использование.

- Симметричное шифрование. Заключается в том, что обе стороны-участники обмена данными имеют абсолютно одинаковые ключи для шифрования и расшифровки данных. Данный способ осуществляет

преобразование, позволяющее предотвратить просмотр информации третьей стороной. Пример: книжный шифр.

- Асимметричное шифрование. Предполагает использовать в паре два разных ключа — открытый и секретный(закрытый). В асимметричном шифровании ключи работают в паре — если данные шифруются открытым ключом, то расшифровать их можно только соответствующим секретным ключом и наоборот — если данные шифруются секретным ключом, то расшифровать их можно только соответствующим открытым ключом. Использовать открытый ключ из одной пары и секретный с другой — невозможно. Каждая пара асимметричных ключей связана математическими зависимостями. Данный способ также нацелен на преобразование информации от просмотра третьей стороной.
- Хеширование. Преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями или функциями свёртки, а их результаты называют хеш-кодом, контрольной суммой или дайджестом сообщения (англ. message digest). Результаты хеширования статистически уникальны. Последовательность, отличающаяся хотя бы одним байтом, не будет преобразована в то же самое значение.

В некоторых странах,есть ограничения на экспорт криптографического программного обеспечения.

США разрешает экспорт программного обеспечения без ограничений, если все следующие пункты выполнены:

- код регулируется экспортными ограничениями ECCN 5D002
- код публично доступен;
- послано уведомление в Бюро промышленности и безопасности США.

Среди свободного программного обеспечения, после выполнения всех оговорённых пунктов, экспорт разрешается для национальных интернет-браузеров и специальных программ, например, TrueCrypt.

3.Частотный анализ

Частотный анализ, частотный криптоанализ — один из методов криптоанализа, основывающийся на предположении о существовании нетривиального статистического распределения отдельных символов и их последовательностей, как в открытом тексте, так и в шифротексте, которое, с точностью до замены символов, будет сохраняться в процессе шифрования и дешифрования.

Упрощённо, частотный анализ предполагает, что частотность появления заданной буквы алфавита в достаточно длинных текстах одна и та же для разных текстов одного языка. При этом, в случае моноалфавитного шифрования, если в шифротексте будет символ с аналогичной вероятностью появления, то можно предположить, что он и является указанной зашифрованной буквой. Аналогичные рассуждения применяются к биграммам (двубуквенным последовательностям), триграммам и т. д. в случае полиалфавитных шифров.

Частотный анализ используется не только в программирование для шифрования анформации. Одним из примеров использования данного метода является статистика.

Объектом исследования в прикладной статистике являются статистические данные, полученные в результате наблюдений или экспериментов. Статистические данные – это совокупность объектов (наблюдений, случаев) и признаков (переменных), их характеризующих. Например, объекты исследования – страны мира и признаки, – географические и экономические показатели их характеризующие: континент; высота местности над уровнем моря; среднегодовая температура; место страны в списке по качеству жизни, доли ВВП на душу населения; расходы общества на здравоохранение, образование, армию; средняя продолжительность жизни; доля безработицы, безграмотных; индекс качества жизни и т.д. Переменные – это величины, которые в результате измерения могут принимать различные значения.

Таблицы частот, или как еще их называют одновходовые таблицы, представляют собой простейший метод анализа категориальных переменных. Таблицы частот могут быть с успехом использованы также для исследования количественных переменных, хотя при этом могут возникнуть трудности с интерпретацией результатов. Данный вид статистического исследования часто используют как одну из процедур разведочного анализа, чтобы посмотреть, каким образом различные группы наблюдений распределены в выборке, или как распределено значение признака на интервале от минимального до максимального значения. Как правило, таблицы частот графически иллюстрируются при помощи гистограмм.

4.Шифрование

Шифрование — обратимое преобразование информации в целях сокрытия от неавторизованных лиц с предоставлением в это же время авторизованным пользователям доступа к ней. Главным образом шифрование служит задачей соблюдения конфиденциальности передаваемой информации.

Пользователи являются авторизованными, если они обладают определённым аутентичным ключом. Вся сложность и, собственно, задача шифрования состоит в том, как именно реализован этот процесс.

В целом, шифрование состоит из двух составляющих: зашифрование и расшифрование.

С помощью шифрования обеспечиваются три состояния безопасности информации:

- Конфиденциальность.

Шифрование используется для скрытия информации от неавторизованных пользователей при передаче или при хранении.

- Целостность.

Шифрование используется для предотвращения изменения информации при передаче или хранении.

- Идентифицируемость.

Шифрование используется для аутентификации источника информации и предотвращения отказа отправителя информации от того факта, что данные были отправлены именно им.

Для того, чтобы прочитать зашифрованную информацию, принимающей стороне необходимы ключ и дешифратор (устройство, реализующее алгоритм расшифровывания). Идея шифрования состоит в том, что злоумышленник, перехватив зашифрованные данные и не имея к ним ключа, не может ни прочитать, ни изменить передаваемую информацию. Кроме того, в современных криптосистемах (с открытым ключом) для шифрования, расшифрования данных могут использоваться разные ключи. Однако, с развитием криптоанализа, появились методики, позволяющие дешифровать закрытый текст без ключа. Они основаны на математическом анализе переданных данных.

5.Шифр Цезаря

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Шифр назван в честь римского полководца Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шифрование с использованием ключа k=3. Буква «Е» «сдвигается» на три буквы вперёд и становится буквой «З». Твёрдый знак, перемещённый на три буквы вперёд, становится буквой «Э», буква «Я», перемещённая на три буквы вперёд, становится буквой «В», и так далее:

Исходный алфавит: А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я
Шифрованный: Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я А Б В

Взлом шифра Цезаря

Шифр Цезаря называют в честь Юлия Цезаря, который, согласно «Жизни двенадцати цезарей» Светония, использовал его со сдвигом 3, чтобы защищать военные сообщения. Хотя Цезарь был первым зафиксированным человеком, использовавшим эту схему, другие шифры подстановки, как известно, использовались и ранее.

“Если у него было что-либо конфиденциальное для передачи, то он записывал это шифром, то есть так изменял порядок букв алфавита, что нельзя было разобрать ни одно слово. Если кто-либо хотел дешифровать его и понять его значение, то он должен был подставлять четвертую букву алфавита, а именно, D, для A, и так далее, с другими буквами.” Гай Светоний Транквилл Жизнь двенадцати цезарей, Книга первая, гл. 56

Шифр Цезаря может быть легко взломан даже в случае, когда взломщик знает только зашифрованный текст. Можно рассмотреть две ситуации:

- Взломщик знает (или предполагает), что использовался простой шифр подстановки, но не знает, что это — схема Цезаря.
- Взломщик знает, что использовался шифр Цезаря, но не знает значение сдвига.

В первом случае шифр может быть взломан, используя те же самые методы что и для простого шифра подстановки, такие как частотный анализ и т. д. Используя эти методы, взломщик, вероятно, быстро заметит регулярность в решении и поймёт, что используемый шифр — это шифр Цезаря.

Во втором случае взлом шифра является даже более простым. Существует не так много вариантов значений сдвига (26 для английского языка), все они могут быть проверены методом грубой силы. Один из способов сделать это — выписать отрывок зашифрованного текста в столбец всех возможных сдвигов — техника, иногда называемая как «завершение простого компонента».

Рассмотрим пример для зашифрованного текста «EXXEGOEXSRGI»; открытый текст немедленно опознается глазом в четвертой строке. Другой способ применения этого метода — это написать алфавит под каждой буквой зашифрованного текста, начиная с этой буквы. Метод может быть ускорен, если использовать заранее подготовленные полоски с алфавитом. Для этого нужно сложить полоски так, чтобы в одной строке образовался зашифрованный текст, тогда в некоторой другой строке мы увидим открытый текст.

Другой подход к применению метода грубой силы для взлома — проверить частотности букв. Изобразив диаграммой частотности букв в зашифрованном тексте, и зная ожидаемое распределение букв для обычного текста на рассматриваемом языке, можно легко определить сдвиг, взглянув на смещение некоторых характерных черт на диаграмме. Этот метод известен как частотный анализ. Например, в тексте на английском языке частотность букв E, T, (обычно наиболее частых), и Q, Z (обычно более редких) особенно различаются. Этот процесс можно автоматизировать, сделав, чтобы компьютерная программа оценивала, насколько хорошо фактическое распределение частотностей соответствует ожидаемому распределению. Например, может использоваться критерий хи-квадрат. Для обычного текста на естественном языке, скорее всего, будет только один вариант декодирования. Но, если использовать очень короткие сообщения, то возможны случаи, когда возможны несколько вариантов расшифровки с различными сдвигами. Например, зашифрованный текст «MPQY» может быть расшифрован как «aden», так и как «know» (предполагая, что открытый текст написан на английском языке). Точно также «ALIIP» можно расшифровать как «dolls» или как «wheel»; «AFCCP» как «jolly» или как «cheer»

Многократное шифрование никак не улучшает стойкость, так как применение шифров со сдвигом a и b эквивалентно применению шифра со сдвигом a + b. В математических терминах шифрование с различными ключами образует группу.

6.Листинг кода

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <iostream>
#include <malloc.h>
#pragma warning(disable : 4996)
#define MAX_LEN 100000

int GetInt(void)
{
	int res = 0;
	int chs = 0;
	do {
		res = scanf_s("%d", &chs);
		while (getchar() != '\n');
		if (!res || chs < 0) printf("Неправильный ввод!\n");
	} while (chs < 0 || res != 1);
	return chs;
}

void Pause()
{
	printf_s("\nДля продолжения нажимите любую клавишу: ");
	_getch();
}

int FreqAnalys()
{
	FILE* fout, *fin;;
	char filename_in[MAX_LEN];
	char filename_out[MAX_LEN];
	int c;
	int* count;
	int i;
	printf("\nВведите имя файла, где нужно провести анализ: ");
```

```
	scanf("%s", filename_in);
	if ((fin = fopen(filename_in, "r")) == NULL)
	{
		printf("\nНе удалось открыть файл. \n ");
		Pause();
		return -1;
	}
	printf("\nВведите имя файла, куда вывести результаты: ");
	scanf("%s", filename_out);
	if ((fout = fopen(filename_out, "w")) == NULL)
	{
		printf("\nНе удалось открыть файл %s на запись. \n ",
filename_out);
		Pause();
		return -1;
	}
	count = (int*)malloc(MAX_LEN * sizeof(int));
	for (i = 0; i < 256; i++)
	{
		count[i] = 0;
	}
	while ((c = getc(fin)) != EOF)++count[c];
	for (i = 64; i <= 255; i++)
	{
		if (count[i] != 0)
		{
			fprintf_s(fout, " %c - %d\n", i, count[i]);
		}
	}
	fclose(fin);
	fclose(fout);
	Pause();
	system("cls");
	return 0;
}
int Encrypt(int n)
{
	FILE* fp1, *fp2;
```

```
char filename_in[MAX_LEN];
int ENG = 26;
int flag;
char c;
printf_s("\nВведите имя файла, который нужно зашифровать: ");
scanf("%s", filename_in);
if ((fp1 = fopen(filename_in, "r")) == NULL)
{
      printf("\nНе удалось открыть файл. \n ");
      Pause();
      return -1;
}
fp2 = fopen("encryptfile.txt", "w");
c = getc(fp1);
while (!feof(fp1))
{
      flag = 0;
      if (c >= 'A' && c <= 'Z')
      {
            c = c + (n % ENG);
            if (c > 'Z') c = 'A' + (c - 'Z');
            fprintf(fp2, "%c", c);
            flag = 1;
      }
      if (c >= 'a' && c <= 'z')
      {
            c = c + (n % ENG);
            if (c > 'z') c = 'a' + (c - 'z');
            fprintf(fp2, "%c", c);
            flag = 1;
      }
      if (!flag) fprintf(fp2, "%c", c);
      c = getc(fp1);
}

fclose(fp1);
fclose(fp2);
```

```
	printf("Расшифровка завершена успешна!\n");
	Pause();
	return 0;
}
int Decrypt(int n)
{
	FILE *fp1, *fp2;
	char filename_in[MAX_LEN];
	int ENG = 26;
	int flag;
	char c;
	printf_s("\nВведите имя файла, который нужно расшифровать: ");
	scanf("%s", filename_in);
	if ((fp1 = fopen(filename_in, "r")) == NULL)
	{
		printf("\nНе удалось открыть файл. \n ");
		Pause();
		return -1;
	}
	if (!(fp2 = fopen("decryptfile.txt", "w")))
	{
		printf("\nНе удалось открыть файл. \n ");
		Pause();
		return -1;
	}
	c = getc(fp1);
	while (!feof(fp1))
	{
		flag = 0;
		if (c >= 'A' && c <= 'Z')
		{
			c = c - (n % ENG);
			if (c < 'A') c = 'Z' - ('A' - c) + 1;
			fprintf(fp2, "%c", c);
			flag = 1;
		}
		if (c >= 'a' && c <= 'z')
```

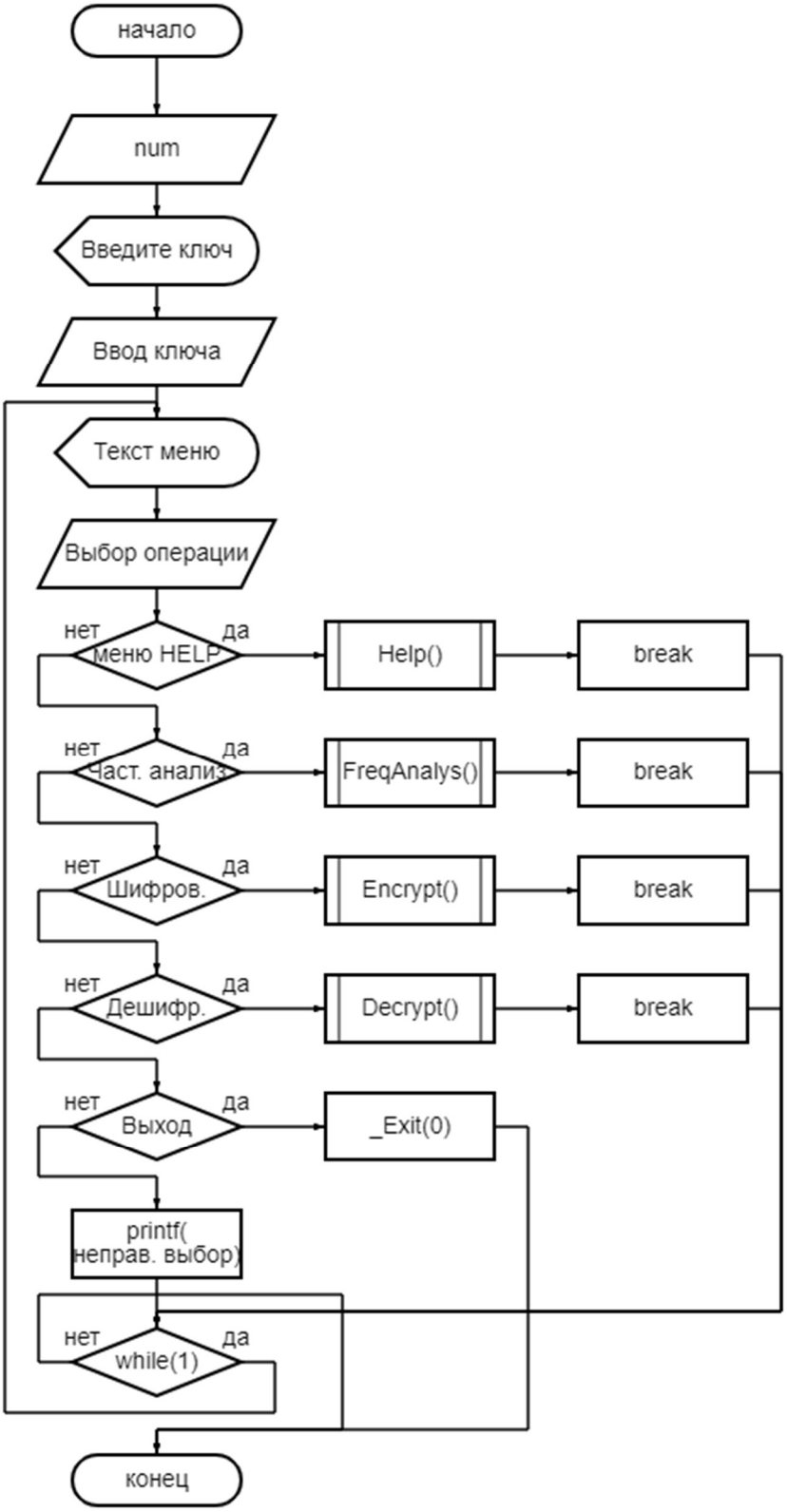
```
            {
                  c = c - (n % ENG);
                  if (c < 'a') c = 'z' - ('a' - c) + 1;
                  fprintf(fp2, "%c", c);
                  flag = 1;
            }
            if (!flag) fprintf(fp2, "%c", c);
            c = getc(fp1);
      }
      fclose(fp1);
      fclose(fp2);
      printf("Расшифровка завершена успешна!\n");
      Pause();
      return 0;
}
void Help()
{
      system("cls");
      printf("\nHelp ");
      printf("\nДля частотного анализа необходимо подготовить файл ");
      printf("\nформата .txt в кодировке Windows.");
      printf("\nДалее следуйте инструкции");
      printf("\nПорядок работы:");
      printf("\n1.Ввести имя файла, откуда будет взять текст для анализа ");
      printf("\n2.Нажать любую клавишу для того, чтобы ");
      printf("\nзакончить работу с функцией");
      printf("\n3.Открыть заданный вами файл вывода и посмотреть результаты");
      printf("\n4.Для дальнейшего использования программы выбирайте нужные вам
функции");
      printf("\nи следуйте инструкциям указанным на экране!");
      printf("\nПриятного пользования!");
      printf("\n© Федорович Илья 150503");
      printf("\n============================================================");
      Pause();
      system("cls");
}
int main(int argc, char** argv)
{
```

```
	system("chcp 1251");
	setlocale(LC_ALL, "rus");
	int num,key;
	system("cls");
	printf("Добро пожаловать в мое приложение!\n");
	printf("Пожалуйства введите ключ который будете использовать в
будущем(от 1 до 26): ");
	key = GetInt();
	while (key > 1 || key > 26)
	{
		printf("Введенный ключ находится вне промежутка 1-26\n");
		printf("Пожалуйста введите ключ еще раз");
		key = GetInt();
	}
	do
	{
		system("cls");
		printf("\nВведите номер функции которой хотите воспользоваться:");
		printf("\n1.Открыть меню HELP");
		printf("\n2.Выполнить частотный анализ текста");
		printf("\n3.Зашифровать текст");
		printf("\n4.Расшифровать текст");
		printf("\n5.Выход из программы\n");
		num = GetInt();
		switch (num)
		{
			case(1):
			{
				rewind(stdin);
				Help();
				break;
			}
			case(2):
			{
				FreqAnalys();
				break;
			}
			case(3):
```

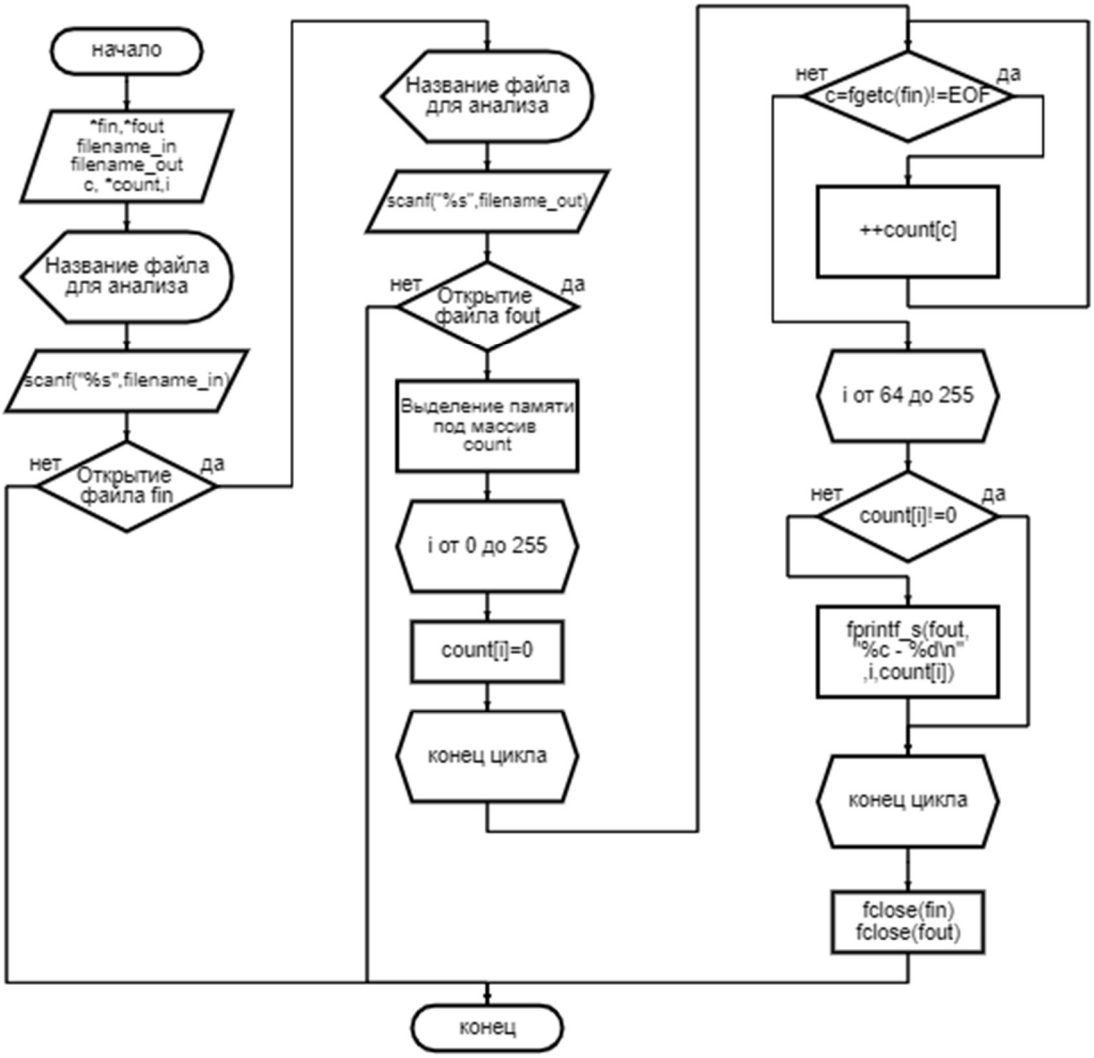
```
                {
                        rewind(stdin);
                        Encrypt(key);
                        break;
                }
                case(4):
                {
                        rewind(stdin);
                        Decrypt(key);
                        break;
                }
                case(5):
                {
                        _Exit(0);
                }
                default:
                {
                        printf("Неправильный выбор!");
                        break;
                }
            }
        } while (1);
}
```

7.Блок-схемы некоторых функций

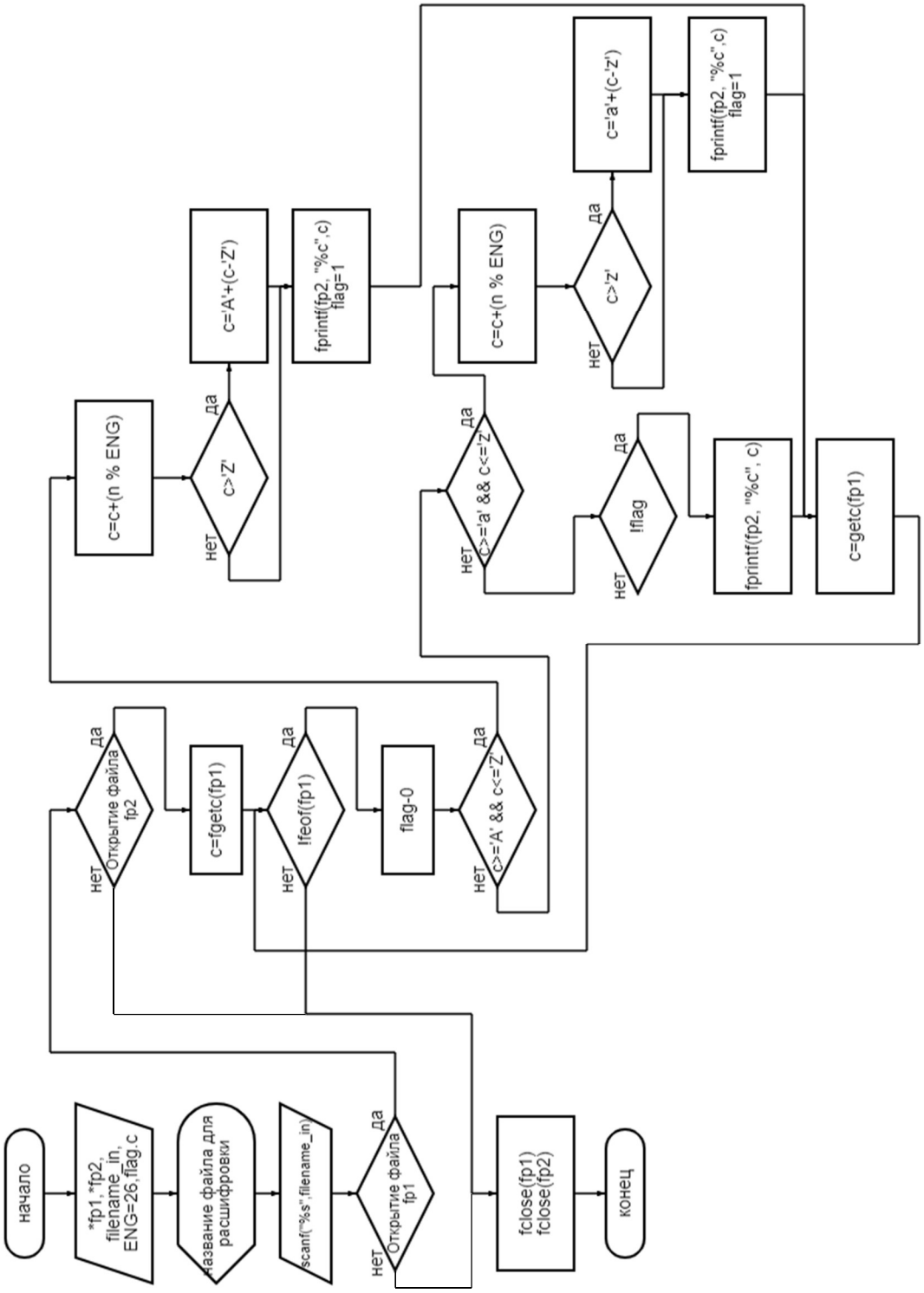
Блок-схема функции main()



Блок-схема функции FriqAnalys()



Блок-схема функции Encrypt()



Заключение

По итогу выполнения учебной практики по дисциплине “Основы алгоритмизации и программирования” была изучена теоретическая информация о основах криптографии, методах шифрования, применены и использованы практические знания. Реализована программа для выполнения операции частотного анализа, шифрования и дешифрования текста.

С каждым днем люди все больше пользуются информацией, передают её, обмениваются знаниями посредством цифровых устройств. Безопасность информация – главное, потому что весь объем информации, находящийся в руках злоумышленников, может привести к печальному последствию.