Московский государственный технический

университет им. Н.Э. Баумана.

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Основы информатики»

Отчет по лабораторной работе № 7

«Шифрование текстовых файлов»

Выполнил:
студент группы ИУ5-13
Терентьев Владислав
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Козлов А. Д.
Подпись и дата:

Москва, 2018 г.

1. Постановка задачи

Написать программу, которая шифрует (шифр Виженера) исходный текст (сообщение) с помощью другого текста (ключа) и записывает в текстовый файл результат, а также выводит статистику для символов исходного текста.

2. Разработка алгоритма

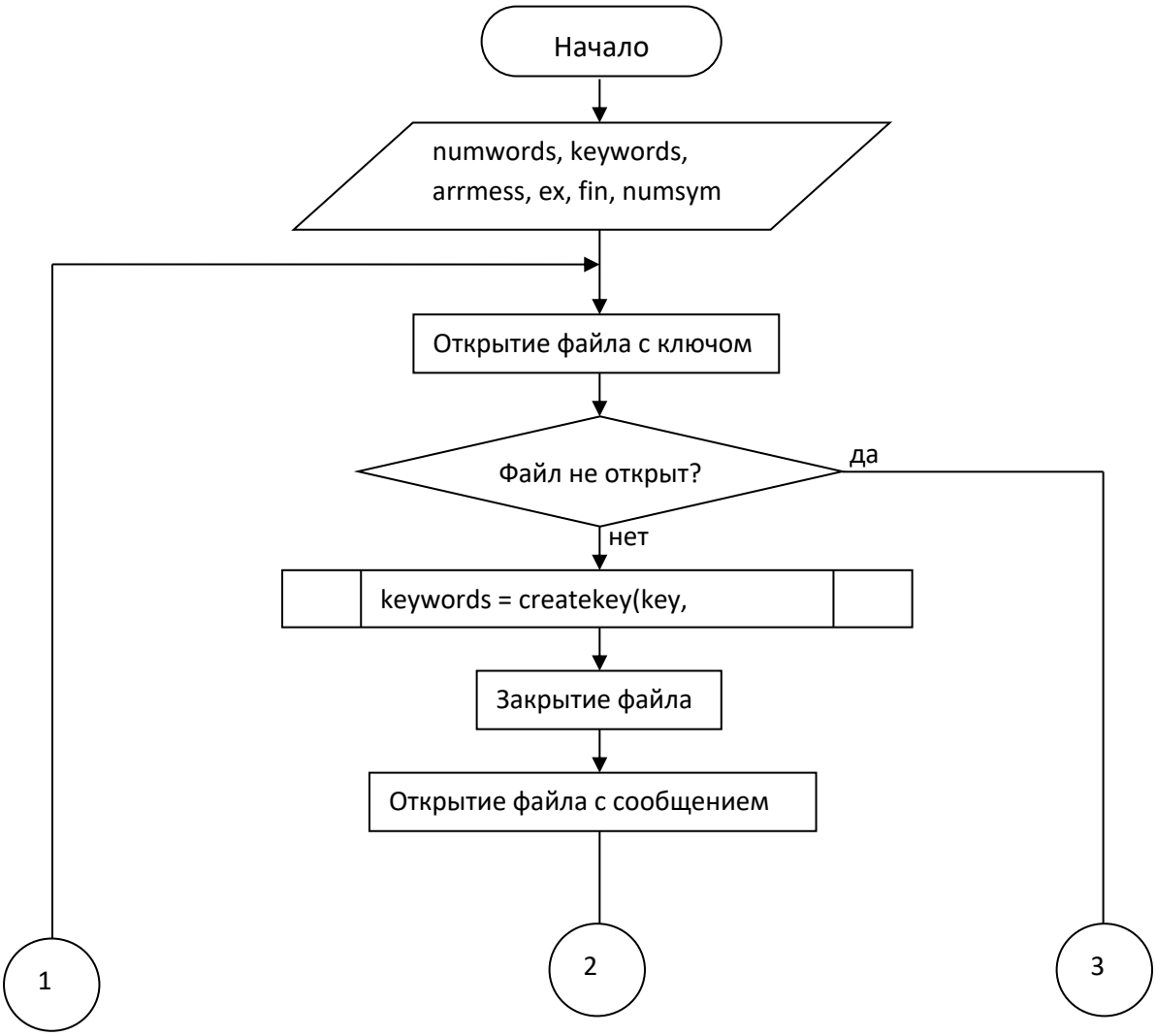
Описание переменных: переменные типа int: numwords – количество слов в ключе; keywords – массив слов ключа, преобразованных в число; arrmess – массив закодированных символов; ex – переменная для цикла; fin – переменная для цикла (в случае сбоя в программе).

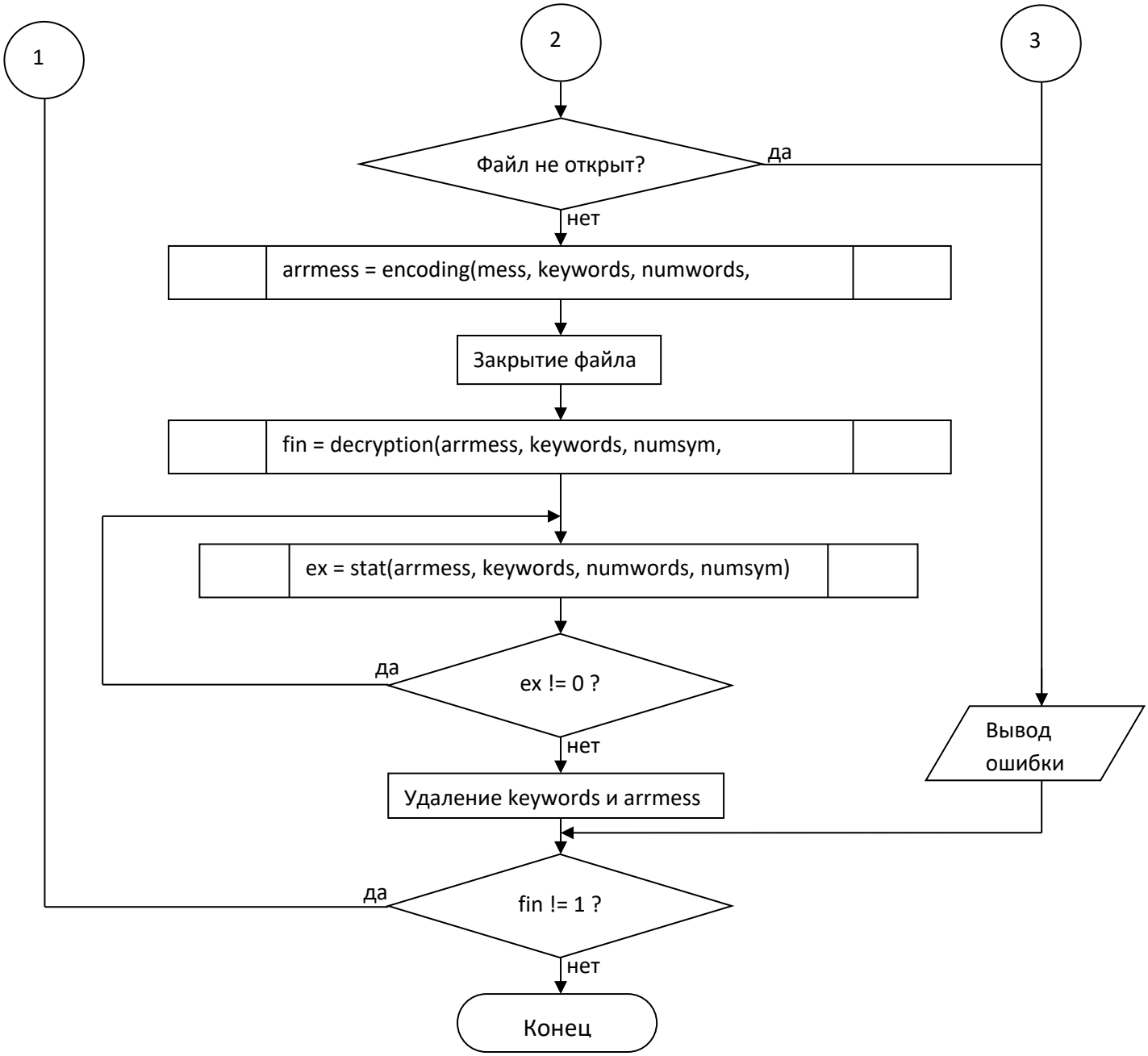
типа long int: numsym – количество символов в сообщении.

Описание функций: функции типа int *: createkey – 2 входных параметра (текстовый документ и адрес переменной), преобразовывает ключ и записывает результат в файл, возвращает keywords; encoding – 4 входных параметра (текстовый документ, массив, число, адрес переменной), шифрует сообщение и записывает результат в два файла (с числами и с символами), возвращает arrmess.

типа int: decryption – 4 входных параметра (2 массива, 2 числа), расшифровывает сообщение и записывает результат в файл, возвращает значение “1”; stat – 4 входных параметра (2 массива, 2 числа), выводит статистику для символа, возвращает значение, введенное пользователем, для повтора выполнения функции.

Схема алгоритма:





3. Текст программы

```
#include "pch.h"
#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

int * createkey(ifstream &, int &);
int * encoding(ifstream &, int *, int, long int &);
int decryption(int *, int *, int, int);
int stat(int *, int *, int, int);

int main()
{
	setlocale(LC_ALL, "Russian");
	int numwords, //количество слов в ключе
		*keywords = NULL, //массив слов ключа, преобразованных в число
		*arrmess = NULL, //массив закодированных символов
		ex = 1, //переменная для цикла
		fin = 0; //переменная для цикла (в случае сбоя в программе)
```

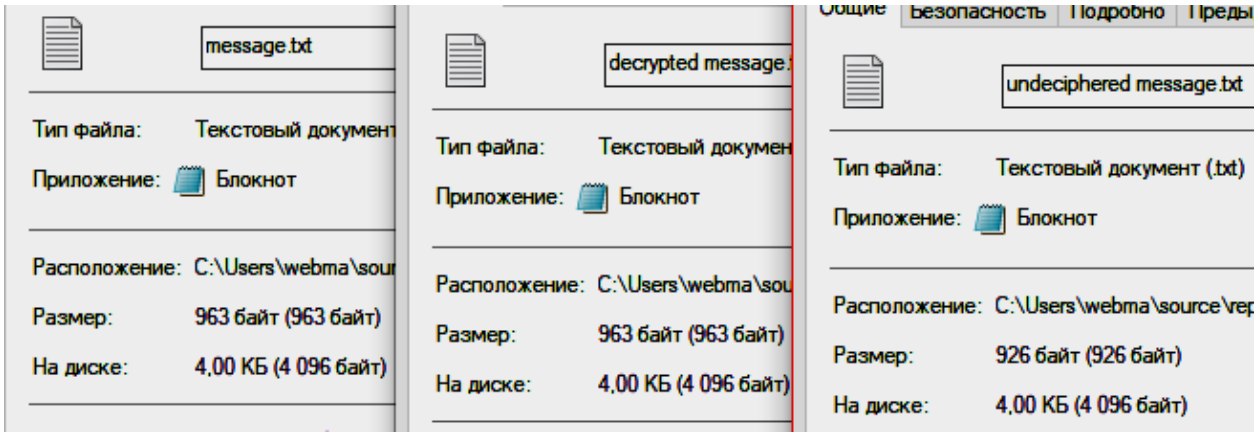
```
	long int numsym;

	do {
		ifstream key("key.txt");
		if (!key.is_open()) {
			cout << "Файл с ключом шифра не может быть открыт! Проверьте
нахождение файла в нужном каталоге и нажмите любую клавишу для повтора." << endl;
			system("pause");
		}
		else
		{
			keywords = createkey(key, numwords);
			key.close();
			ifstream mess("message.txt");
			if (!mess.is_open()) {
				cout << "Файл с текстом для шифровки не может быть открыт!
Проверьте нахождение файла в нужном каталоге и нажмите любую клавишу для повтора." <<
endl;
				system("pause");
			}
			else {
				arrmess = encoding(mess, keywords, numwords, numsym);
				mess.close();
				fin = decryption(arrmess, keywords, numsym, numwords);
				do {
					ex = stat(arrmess, keywords, numwords, numsym);
				} while (ex != 0);
				delete[] keywords;
				delete[] arrmess;
			}
		}
	} while (fin != 1);
	return 0;
}

int * createkey(ifstream &key, int &numwords) {
	key.seekg(0, ios::end);
	long int numsymex = key.tellg();
	key.seekg(0, ios::beg);
	int *arrkey = new int[numsymex];
	cout << "Ключ: " << endl << endl;
	long int numsym = 0;
	char sym;
	numwords = 0;
	while (!key.eof()) {
		key.get(sym);
		if ((sym == ' ') || (int(sym) == 10)) {
			numwords++;
		}
		unsigned char symex = sym;
		arrkey[numsym] = int(symex);
		numsym++;
	}
	numwords = numwords + 1;

	int *keywords = new int[numwords];
	int i;
	for (i = 0; i < numwords; i++) {
		keywords[i] = 0;
	}

	ofstream kword("words code.txt", ios_base::out | ios_base::trunc);
	int nmword = 0;
	for (i = 0; i < numsym - 1; i++) {
		cout << unsigned char(arrkey[i]);
```

```
			if ((arrkey[i] == 32) || (arrkey[i] == 10)) {
				keywords[nmword] = keywords[nmword] % 256;
				kword << keywords[nmword] << " ";
				nmword++;
			}
			else {
				keywords[nmword] = keywords[nmword] + arrkey[i];
			}
		}
		keywords[nmword] = keywords[nmword] % 256;
		kword << keywords[nmword];
		kword.close();
		cout << endl << endl << "Кодовые слова записаны в файл 'words code.txt'" << endl;
		delete[] arrkey;
		return keywords;
}

int * encoding(ifstream &mess, int *keywords, int numwords, long int &numsym) {
		mess.seekg(0, ios::end);
		long int numsymex = mess.tellg();
		mess.seekg(0, ios::beg);
		int *arrmess = new int[numsymex];
		cout << endl << "Шифруемый текст: " << endl << endl;
		numsym = 0;
		char sym;
		while (!mess.eof()) {
			mess.get(sym);
			unsigned char symex = sym;
			arrmess[numsym] = int(symex);
			numsym++;
		}
		ofstream cryp("cipher.txt", ios_base::out | ios_base::trunc);
		int i;
		for (i = 0; i < numsym - 1; i++) {
			cout << unsigned char(arrmess[i]);
			arrmess[i] = (arrmess[i] + keywords[i%numwords]) % 256;
			cryp << arrmess[i] << " ";
		}
		cryp.close();
		cout << endl << endl << "Зашифрованный текст записан в файл 'cipher.txt'" << endl
<< endl;
		ofstream symcryp("undeciphered message.txt", ios_base::out | ios_base::trunc);
		for (i = 0; i < numsym - 1; i++) {
			symcryp << unsigned char(arrmess[i]);
		}
		symcryp.close();
		cout << "Нерасшифрованный текст записан в файл 'undeciphered message.txt'" << endl
<< endl;
		return arrmess;
}

int decryption(int *arrmess, int *keywords, int numsym, int numwords) {
		ofstream decryp("decrypted message.txt", ios_base::out | ios_base::trunc);
		int i;
		for (i = 0; i < numsym - 1; i++) {
			arrmess[i] = arrmess[i] - keywords[i%numwords];
			if (arrmess[i] < 0) {
				arrmess[i] = arrmess[i] + 256;
			}
			decryp << unsigned char(arrmess[i]);
		}
		decryp.close();
		cout << "Расшифрованный текст записан в файл 'decrypted message.txt'" << endl;
		return 1;
}
```

```
int stat(int *arrmess, int *keywords, int numwords, int numsym) {
	cout << endl << "Введите символ, статистику которого вы хотите узнать: " << endl;
	unsigned char sym;
	sym = cin.get();
	cin.get();
	int isym = int(sym);
	if (isym > 127) {
		isym = isym + 64;
	}
	int *msym = new int[numwords];
	int i;
	for (i = 0; i < numwords; i++) {
		msym[i] = 0;
	}
	int inw,
		sumsym = 0,
		symmax = 0,
		symmin;
	cout << endl << "Данный символ кодируется следущими кодами: " << endl;
	int *mtr = new int[257];
	for (i = 0; i < 257; i++) {
		mtr[i] = 0;
	}
	for (i = 0; i < numsym - 1; i++) {
		if (arrmess[i] == isym) {
			inw = i % numwords;
			msym[inw]++;
			cout << (arrmess[i] + keywords[inw]) % 256 << " ";
			mtr[(arrmess[i] + keywords[inw]) % 256]++;
			sumsym++;
			if (msym[inw] > symmax) {
				symmax = msym[inw];
			}
		}
	}
	cout << endl << endl << "Символ в шифре, код символа и количество:" << endl;

	for (i = 0; i < 257; i++) {
		if (mtr[i] != 0) {
			cout << unsigned char(i) << setw(8) << i << setw(8) << mtr[i] <<
endl;
		}
	}
	symmin = msym[0];
	for (i = 1; i < numwords; i++) {
		if (msym[i] < symmin) {
			symmin = msym[i];
		}
	}
	double mid = double(sumsym) / numwords;
	cout << endl << "В среднем данный элемент попадает на одно кодовое слово " << mid
<< " раз(а)." << endl << endl;
	double symmax2 = (symmax - mid) / (mid / 100);
	double symmin2 = (mid - symmin) / (mid / 100);
	if (symmax2 > symmin2) {
		cout << "Значение максимального отклонения от нормы: " << symmax << ". В
процентах: " << symmax2 << " %.";
	}
	else {
		cout << "Значение максимального отклонения от нормы: " << symmin << ". В
процентах: " << symmin2 << " %.";
	}
	int ex;
```

```
	cout << endl << endl << "Вы хотите узнать статистику другого элемента? (1 - да, 0
- нет)" << endl;
	cin >> ex;
	cin.get();
	sumsym = 0;
	symmax = 0;
	delete[] msym;
	return ex;
}
```

4. Анализ результатов



Размеры исходного и расшифрованного текста совпадают, но размер нерасшифрованного текста в символах не совпадает. Это вызвано тем, что при шифровании создался символ, код которого в ASCII меньше чем 32, это означает что этот символ не отображается в текстовом документе и не имеет размера.

```
Ключ:

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very
 much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just did
n't hold with such nonsense.
Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any n
eck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amou
nt of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neig
hbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.

Кодовые слова записаны в файл 'words code.txt'

Шифруемый текст:

Песенка разбойников:

1. Говорят, мы бяки-буки,
Как выносит нас земля?
Дайте что ли карты в руки
Погадать на короля.
Ой-ля-ля, ой-ля-ля,
Погадать на короля,
Ой-ля-ля, ой-ля-ля,
Ех-ха!

2. Завтра дальняя дорога
Выпадает королю.
У него деньжонок много,
А я денежки люблю.
Ой-лю-лю, ой-лю-лю,
А я денежки люблю.
Ой-лю-лю, ой-лю-лю,
Ех-ха!

3. Королева карта бита,
Бит и весь его отряд.
Дело будет шито-крыто -
Карты правду говорят.
Ой-ля-ля, ой-ля-ля,
Завтра грабим короля.
Ой-ля-ля, ой-ля-ля,
Ех-ха!

Ничего на свете лучше нету

Ничего на свете лучше нету,
Чем бродить друзьям по белу свету.
Тем, кто дружен, не страшны тревоги,
Нам любые дороги дороги.

Наш ковер - цветочная поляна.
Наши стены - сосны великаны.
Наша крыша - небо голубое,
Наше счастье - жить такой судьбою.

Мы свое призванье не забудем:
Смех и радость мы приносим людям.
Нам дворцов заманчивые своды
Не заменят никогда свободы.

Зашифрованный текст записан в файл 'cipher.txt'

Нерасшифрованный текст записан в файл 'undeciphered message.txt'

Расшифрованный текст записан в файл 'decrypted message.txt'

Введите символ, статистику которого вы хотите узнать:
о

Данный символ кодируется следущими кодами:
59 163 115 119 133 98 195 73 79 33 159 162 214 161 33 121 208 195 15 85 197 203 56 197 180 46 208 33 2 168 162 85 195
 156 187 130 46 73 20 126 114 33 195 203 59 168 75 201 197 161 195 146 45 170 203 214 168 47 57 146 195 132 136 203 2
 119

Символ в шифре, код символа и количество:
□       2       2
□      15       1
□      20       1
!      33       4
-      45       1
.      46       2
/      47       1
8      56       1
9      57       1
;      59       2
I      73       2
K      75       1
O      79       1
U      85       2
b      98       1
r     114       1
s     115       1
w     119       2
y     121       1
~     126       1
'     130       1
"     132       1
:     133       1
?     136       1
'     146       2
?     156       1
?     159       1
Ў     161       2
ў     162       2
?     163       1
Ё     168       3
є     170       1
?     180       1
»     187       1
Г     195       6
Е     197       3
Й     201       1
Л     203       4
Р     208       2
Ц     214       2

В среднем данный элемент попадает на одно кодовое слово 0.507692 раз(а).

Значение максимального отклонения от нормы: 3. В процентах: 490.909 %.

Вы хотите узнать статистику другого элемента? (1 - да, 0 - нет)
```