|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт комплексной безопасности и специального приборостроения | | |
| Кафедра КБ-1 «Защита информации» | | |

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ**

|  |
| --- |
| *(вид практики)*  **ПРАКТИКУ ПО ПОЛУЧЕНИЮ ПЕРВИЧНЫХ ЗНАНИЙ И УМЕНИЙ**  *(тип практики)* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **студентам** | 1 | **курса учебной группы** | БББО-05-20 |  |
| Муханову Матвею Эдуардовичу, Кутьину Захару Сергеевичу | | | | |
| *(фамилия, имя и отчество)* | | | | |

|  |  |
| --- | --- |
| **Место и время практики:** | Институт комплексной безопасности и специального приборостроения, кафедра КБ-1 «Защита информации», с 08.02.2021 по 01.06.2021 |
| **Должность на практике:** | Техник |
| **1. ЦЕЛЕВАЯ УСТАНОВКА:** | Получение первичных профессиональных умений и опыта профессиональной деятельности |
| **2. СОДЕРЖАНИЕ ПРАКТИКИ:** |  |
| 2.1. Изучить: | Изучить материалы по дисциплине «Основы информационной безопасности». |
| 2.2. Практически выполнить: | Поиск требуемой информации в сети Интернет, заполнить отчет о практике, выполнить индивидуальное задание |
| 2.3. Ознакомиться: | Ознакомиться с работой основных комплексов контроля защищенности (аудита информационной безопасности) кафедры КБ-1 «Защита информации» |
| **3.ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:** | «Программа, реализующая алгоритм строгой двухфакторной аутентификации на основе случайных чисел с использованием шифра Хилла» |
| **4.ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:** | При изучении материалов по дисциплине «Основы информационной безопасности» обратить особое внимание на Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" от 26.07.2017 N 187-ФЗ (последняя редакция) |

И.О. Заведующий кафедрой:

|  |  |  |  |
| --- | --- | --- | --- |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. |  |  | Трубиенко О.В. |
|  | *Подпись* |  | *ФИО* |

**СОГЛАСОВАНО:**

Руководитель практики от кафедры:

|  |  |  |  |
| --- | --- | --- | --- |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. |  |  | Головченко Д.А. |
|  | *Подпись* |  | *ФИО* |

Задание получили:

|  |  |  |  |
| --- | --- | --- | --- |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. |  |  | Муханов М.Э. |
|  | *Подпись* |  | *ФИО* |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. |  |  | Кутьин З.С. |
|  | *Подпись* |  | *ФИО* |

**Проведенные инструктажи:**

Охрана труда: «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г.

Инструктирующий ТрубиенкоО.В. , доцент

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка, должность* |

Инструктируемый Муханов М.Э.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

Инструктируемый Кутьин З.С.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

Техника безопасности: «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г.

Инструктирующий Трубиенко О.В. , доцент

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка, должность* |

Инструктируемый Муханов М.Э.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

Инструктируемый Кутьин З.С.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

Пожарная безопасность: «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г.

Инструктирующий Трубиенко О.В. , доцент

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка, должность* |

Инструктируемый Муханов М.Э.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

Инструктируемый Кутьин З.С.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Подпись* |  | *Расшифровка* |

С правилами внутреннего распорядка ознакомлен: «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Муханов М.Э. |
|  | *Подпись* |  | *Расшифровка* |

С правилами внутреннего распорядка ознакомлен: «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Кутьин З.С. |
|  | *Подпись* |  | *Расшифровка* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
| МИНОБРНАУКИ РОССИИ | | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | |
| Институт комплексной безопасности и специального приборостроения | | | |
| Кафедра КБ-1 «Защита информации» | | | |
| **ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**  *(вид практики)*  **ПРАКТИКЕ ПО ПОЛУЧЕНИЮ ПЕРВИЧНЫХ ЗНАНИЙ И УМЕНИЙ**  *(тип практики)* | | | |
| **Тема практики:** | Написание программы, реализующей протокол строгой двусторонней аутентификации на основе случайных чисел с использованием шифра Хилла | | |
|  |  | | |
| приказ Университета о направлении на практику от «08» февраля 2021г. № 290-У | | | |

**Отчет представлен к**

**рассмотрению:**

Студенты группы БББО-05-20 «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_ г. Муханов М.Э.

(подпись и расшифровка подписи)

«\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_ г. Кутьин З.С.

(подпись и расшифровка подписи)

**Отчет утвержден.**

**Допущен к защите:**

Руководитель практики от кафедры: «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_ г. Головченко Д.А.

(подпись и расшифровка подписи)

Москва 2021

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт комплексной безопасности и специального приборостроения | | |
| Кафедра КБ-1 «Защита информации» | | |

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ**

**УЧЕБНОЙ ПРАКТИКИ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| студентов | | Муханова Матвея Эдуардовича, Кутьина Захара Сергеевича | | | | | | | | | |  | 1 | курса |
|  | | *Фамилия Имя Отчество* | | | | | | | | | |  |  |  |
| группы | БББО-05-20 | | | , | очной | | | | формы обучения, обучающихся | | | | | |
|  |  | | |  | *Очной, очно-заочной, заочной* | | | |  | | | | | |
| по направлению подготовки/специальности | | | | | | | 10.03.01 | | | | | | | |
|  | | | | | | | | *Код направления подготовки/специальности* | | | | | | |
| Информационная безопасность | | | | | | | | | | | | | | |
| *Наименование направления подготовки/специальности* | | | | | | | | | | | | | | |
| профиль/магистерская программа/специализация | | | | | | | | | | Безопасность автоматизированных систем | | | | |
|  | | | | | | | | | |  | | | | |
| **Неделя** | | | **Сроки выполнения** | | | **Этап** | | | | | **Отметка о выполнении** | | | |
| 1 | | | 08.02.2021 г. | | | Инструктаж по требованиям техники безопасности | | | | |  | | | |
| 2 | | | 15.02.2021 г. | | | Выбор темы на проектно-технологическую практику | | | | |  | | | |
| 3 | | | 25.02.2021 г. | | | Консультация с руководителем по выбранной теме | | | | |  | | | |
| 4 | | | 04.03.2021 г. | | | Обзор литературы на выбранную тему | | | | |  | | | |
| 5 | | | 11.03.2021 г. | | | Разработка плана практической работы | | | | |  | | | |
| **Неделя** | | | **Сроки выполнения** | | | **Этап** | | | | | **Отметка о выполнении** | | | |
| 6 | | | 18.03.2021г. | | | Поиск необходимой информации о шифре Хилла | | | | |  | | | |
| 7 | | | 25.03.2021г. | | | Обзор литературы о шифре Хилла | | | | |  | | | |
| 8 | | | 01.04.2021г. | | | Анализ найденной информации о шифре Хилла | | | | |  | | | |
| 9 | | | 08.04.2021г. | | | Поиск необходимой информации о методах шифра Хилла | | | | |  | | | |
| 10 | | | 15.04.2021г. | | | Промежуточный контроль проделанной работы | | | | |  | | | |
| 11 | | | 22.04.2021г. | | | Обзор литературы о методах шифра Хилла | | | | |  | | | |
| 12 | | | 29.04.2021г. | | | Исследование основных методов реализации шифра Хилла | | | | |  | | | |
| 13 | | | 13.05.2021г. | | | Анализ основных методов реализации шифра Хилла | | | | |  | | | |
| 14 | | | 20.05.2021г. | | | Работа над отчетом | | | | |  | | | |
| 15 | | | 27.05.2021г. | | | Подготовка к защите отчета по практике | | | | |  | | | |
| 16 | | | 01.06.2021 г. | | | Защита отчета по практике | | | | |  | | | |

|  |  |  |
| --- | --- | --- |
| **СОГЛАСОВАНО:** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| И.О. Заведующий кафедрой |  |  | Доцент, Пушкин П.Ю. |
|  | *Подпись* |  | *Ученая степень, ученое звание, должность, ФИО* |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель практики от кафедры |  |  | Головченко Д.А. |
|  | *Подпись* |  | *Ученая степень, ученое звание, должность, ФИО* |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся |  |  | Муханов М.Э. |
|  | *Подпись* |  | *ФИО* |
| Обучающийся |  |  | Кутьин З.С. |
|  | *Подпись* |  | *ФИО* |

**ОТЧЁТ**

**по учебной практике**

*(вид практики)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **студентов** | | 1 | **курса учебной группы** | БББО-05-20 |  |
| Муханова Матвея Эдуардович, Кутьина Захара Сергеевича | | | | | |
| *(фамилия, имя и отчество)* | | | | | |
| **Института/Филиала** | Институт комплексной безопасности и специального приборостроения | | | | |
|  | *(наименование института/филиала полностью)* | | | | |

1. Практику проходили «08» февраля 2021 г. по «01» июня 2021 г.

в Институте комплексной безопасности и специального приборостроения, кафедра КБ-1

|  |
| --- |
| «Защита информации», техники |
| *(место прохождения практики и должность)* |

2. Задание на практику выполнили

|  |
| --- |
| В полном объеме |
| *(указать: в полном объеме или частично)* |

* 1. Не выполнены следующие задания:

|  |
| --- |
| - |
| *(указать также причины невыполнения)* |

3. Подробное содержание выполненной на практике работы и достигнутые результаты:

|  |
| --- |
| Получены профессиональные навыки и умения. Изучены материалы по дисциплине |
| «Основы информационной безопасности». Изучены основные комплексы контроля |
| защищенности кафедры КБ-1 «Защита информации». Выполнено дополнительное |
| задание, изучена тема «Шифр Хилла». |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Муханов М.Э. |
|  | *(подпись)* |  | *(фамилия и инициалы)* |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. | | | | |
| Студент |  |  | Кутьин З.С. |
|  | *(подпись)* |  | *(фамилия и инициалы)* |
| «\_\_\_»  \_\_\_\_\_\_\_\_\_\_\_\_  20\_\_\_ г. | | | | |

**Отчет проверил:**

**Руководитель практики от Университета**

|  |  |  |
| --- | --- | --- |
|  |  | Головченко Д.А. |
| *(подпись)* |  | *(фамилия и инициалы)* |

Оглавление

[Введение 9](#_Toc74261262)

[Основная часть 10](#_Toc74261263)

[1. История шифра Хилла 10](#_Toc74261264)

[2. Описание шифра Хилла 10](#_Toc74261265)

[3. Написание программы 12](#_Toc74261266)

[Полный код программы указан в приложении в разделе листинг. 15](#_Toc74261267)

[4. Проверка программы на наличие недоработок и на работу с матрицами разных размеров. 16](#_Toc74261268)

[Заключение 17](#_Toc74261269)

[Список используемой литературы 18](#_Toc74261270)

[Приложение 19](#_Toc74261271)

[1. Рисунки 19](#_Toc74261272)

[2. Листинг 25](#_Toc74261273)

# Введение

Цель проекта: разработать программу, реализующую протокол строгой двусторонней аутентификации на основе случайных чисел с использованием Шифра Хилла на языке C++.

Задачи проекта:

1. Изучить необходимую литературу и статьи об алгоритме шифрования Хилла.
2. Изучить необходимую литературу и статьи о выбранном языке программирования (C++).
3. Разработать и написать программу, реализующую протокол строгой двусторонней аутентификации на основе случайных чисел с использованием шифра Хилла.
4. Проанализировать программу на наличие недоработок; проверить работу программы с матрицами разных размерностей.
5. Сделать выводы о теме и проделанной работе, о возможном использовании полученных навыков и умений, а также написанной программы.

Актуальность: в настоящее время активно развиваются информационные технологии, и роль информации важна как никогда. Поэтому защита сведений и персональных данных стала неотъемлемой частью повседневной жизни людей. Тема данного проекта связана с безопасностью информации обеспечением ее целостности, доступности и конфиденциальности, что говорит об актуальности работы.

# Основная часть

## История шифра Хилла

Названный по фамилии своего создателя, американского математика Лестера Хилла, данный шифр представляет собой способ шифрования информации на основе линейной алгебры и модульной арифметики, созданный в 1929 году. И хоть он не нашёл практического применения в криптографии из-за слабой устойчивости к взлому, а также отсутствия методик генерации матриц большого размера, но, как описал его криптограф Дэвид Кан в своей книге “Взломщики кодов”:

“*Хилл был одним из тех, кто разработал общий и мощный метод. К тому же шифр Хилла впервые перевёл криптографию с использованием полиграмм в разряд практических дисциплин*”.

## Описание шифра Хилла

Шифр Хилла – полиграммный шифр, то есть в нём исходные элементы заменяются зашифрованным текстом в соответствии с определённым правилом. Для шифрования текста, необходимо проделать следующие шаги:

* 1. Задать кодированный алфавит. Так, для шифрования строчных и заглавных символов русского алфавита, десяти цифр и семи специальных символов будет использоваться алфавит длиной в 83 символа. Каждому символу алфавита в соответствие ставится целочисленный код, например, нумерация от 0 до 82 в описанном выше случае.
  2. Следующим шагом берётся текст, каждый символ которого заменяется на код, соответствующий определенному символу алфавита.
  3. На этом этапе необходимо задать квадратную ключ-матрицу с элементами – кодами нашего алфавита.
  4. Далее закодированный по алфавиту исходный текст нужно разбить на блоки длиной равной размерности ключ-матрицы. В случае, если в последнем блоке недостаёт элементов, следует дополнить его.
  5. Теперь каждый блок умножается на ключ-матрицу, а после каждый элемент полученной матрицы делится по остатку на число равное длине алфавита.
  6. Теперь полученные матрицы необходимо декодировать с помощью алфавита. Полученные блоки “склеиваем”. Это и есть зашифрованное сообщение.

Для дешифрования нужно проделать следующие шаги:

* 1. Зашифрованный текст обратно кодируем с помощью алфавита.
  2. Находим определитель ключ-матрицы.
  3. По расширенному алгоритму Евклида находим d, x, y.
  4. На этом этапе нужно найти обратный детерминанту элемент в кольце по модулю длины нашего алфавита. Для этого:
     + Если детерминант и x – положительные, то обратный детерминанту элемент будет равен x.
     + Если детерминант и x – отрицательные, то обратный детерминанту элемент будет равен –x.
     + Если детерминант отрицательный, а x – положительный, то обратный детерминанту элемент будет равен x.
     + Если детерминант положительный и x – отрицательный, то обратный детерминанту элемент будет равен x + (длина алфавита).
  5. Теперь находим матрицу обратную ключ-матрице по модулю длины алфавита.
  6. Последним шагом разбираем кодированный текст на блоки и умножаем каждый блок на матрицу обратную матрице-ключу. Каждый элемент полученных блоков делим по модулю на длину алфавита и декодируем по алфавиту. Полученные блоки “склеиваем”. Это и есть дешифрованное сообщение.

## Написание программы

Программа, реализующая алгоритм, написана на языке C++ в среде разработки Microsoft Visual Studio 2019.

В программе присутствует ограничение выбора размерности ключ-матрицы: от матриц 1x1 до матриц 6x6. Это ограничение обусловлено тем, что используемые программные средства не позволяют вместить в себя больший объём данных, т.е. при перемножении матриц размером свыше, чем 6x6 будет происходить переполнение буфера. Избежать данной проблемы можно, если использовать нестандартные типы данных языка C++, рассчитанные на большее количество байт, в отличие от типа данных long long, используемого в программе.

Описание программы:

С самого начала указаны библиотеки, необходимые для дальнейшей корректной работы программы (рис. 1):

* 1. #include <iostream> - подключение через директиву препроцессора для подключения файлов #include файла библиотеки iostream, необходимой для организации ввода-вывода.
  2. #include <windows.h>- подключение файла заголовков, содержащего заявления для всех функций в Windows API, все общие макросы, и все типы данных, используемых различными функциями и подсистем. Также эта библиотека позволяет подключить в консоли русификацию.
  3. #include <string> - подключение заголовочного файла для организации работы со строками.
  4. #include <math.h>– подключение заголовочного файла для использования специальных математических функций языка программирования C++ (стандартная математическая библиотека).
  5. #include <ctime> – подключение библиотеки ctime, необходимой для преобразования значения типа time\_t в строку (нужна для корректного использования генерации случайных чисел).

Далее прописываются (рис. 2):

* 1. #define – директивы препроцессора, определяющие идентификатор и последовательность символом, на которые будет заменён данный идентификатор при обнаружении в тексте программы.
  2. char abc[SIZE] – массив с символьным типом данных char. В каждой ячейке массива хранится символ. В данный массив записываются все символы выбранного нами алфавита.
  3. using namespace std – директива using позволяет использовать все имена в namespace для пространства имён std без имени пространства имён в качестве явного квалификатора.

Следующим шагом были реализованы функции, необходимые для работы программы и шифрования\дешифрования строки символов (рис. 3а, 3б, 3в, 3г, 3д, 3е, 3ж).

|  |  |
| --- | --- |
| Перечень используемых функций | |
| Функция | Какую задачу реализует |
| void setData() | Заполняет квадратную матрицу размерности k случайными числами, не превышающими мощность алфавита с помощью стандартной функции генерации случайных чисел. |
| void getMatrix() | Выполняет вывод на экран квадратной матрицы размерностью k. |
| void getAbc() | Выполняет вывод на экран элементов заданного алфавита. |
| string alphabetCheck() | Проверяет символы строки на наличие символов, не принадлежащих заданному алфавиту. В случае, если такой символ найден – произойдет вывод сообщения об ошибке и выполнится запрос нового ввода строки. |
| longlongfindDet() | Находит детерминант квадратной матрицы размерностью n. |
| void phraseAddition() | Дополняет строку “пробелами” до длины, кратной размерности матрицы. |
| void phraseToNumbers() | Представляет строку в виде кода в соответствии с индексами алфавита. |
| void encrypt() | Получает на вход от функции void cryptHill() блоки входной строки, после чего выполняет умножение данного блока на ключ-матрицу. |
| void cryptHill() | Выполняет шифрование фразы, уже представленной в виде кода, согласно алфавиту, путём разбиения на блоки по размерности ключ-матрицы и следующей отправкой блоков в функцию void encrypt(). |
| long long evklid() | По расширенному алгоритму Евклида находит X, необходимый для вычислений обратного детерминанту элемента в кольце по модулю размерности алфавита. |
| long long findDetobr() | Вычисляет обратный детерминанту элемент в кольце по модулю длины алфавита. |
| void algCompl() | Находит алгебраические дополнения для вычисления обратной матрицы ключа по модулю длины алфавита. |
| void transpose() | Транспонирует матрицу. |
| void posElem() | Добавляет к отрицательным элементам матрицы число, равное длине выбранного алфавита. |
| void Decrypt() | Функция дешифрования. Выполняет последовательный вызов функций void algCompl(), void transpose(), void posElem(). |

## Полный код программы указан в приложении в разделе листинг.

|  |  |
| --- | --- |
| Перечень используемых переменных | |
| Переменная | Для чего предназначена |
| long long key | Переменная для хранения размерности матрицы-ключа. |
| long long\*\* matrix | Двумерный динамический массив символов для хранения матрицы-ключа. |
| long long\*\* matrix1 | Двумерный динамический массив символов для хранения обратной матрицы к матрице-ключу при первом цикле дешифрования. |
| long long\*\* matrix2 | Двумерный динамический массив символов для хранения обратной матрицы к матрице-ключу при втором цикле дешифрования. |
| long long det | Переменная для хранения значения детерминанта матрицы-ключа. |
| string phrase | Строка, которая хранит введенную пользователем для дальнейшего шифрования. |
| long long\* word | Одномерный динамический массив для хранения фразы, представленной в виде кода в соответствие с индексами алфавита. |
| long long\* word1 | Одномерный динамический массив для хранения блоков по key символов кода. |
| long long\* resMatrix | Одномерный динамический массив для хранения зашифрованной один раз фразы, представленной в виде кода. |
| long long x, y, X | Переменные необходимые для вычисления обратного детерминанту элемента в кольце по модулю размерности алфавита. |
| long long detobr | Переменная для хранения обратного детерминанту элемента в кольце по модулю длины алфавита. |
| long long\* newMatr | Одномерный динамический массив для хранения зашифрованной второй раз фразы, представленной в виде кода. |

## Проверка программы на наличие недоработок и на работу с матрицами разных размеров.

Во время проверки не было выявлено недоработок, программа исправно работает в различных исключительных ситуациях: при задании пользователем размерности не из диапазона [1;6], а также при вводе фразы, включающей символы, не принадлежащие алфавиту. При данных исключительных ситуациях пользователю выводится сообщение об ошибке и необходимости повторного ввода данных (рис. 4а, 4б).

При тестировании программы с размерностями ключ-матриц из диапазона [1;6] не было обнаружено ошибок (рис. 5).

# Заключение

По завершению работы, можно сделать следующие выводы:

1. Была изучена необходимая литература и Интернет-статьи об алгоритме шифрования Хилла: истории шифра и принципах его работы.
2. Были изучены книги и статьи, которые помогли в понимании языка программирования C++ для написания будущего кода программы.
3. Была разработана программа, корректно реализующая протокол строгой двусторонней аутентификации на основе случайных чисел с использованием шифра Хилла.
4. Программа была проверена на наличие неисправностей. Она успешно выполняет поставленную задачу: правильно шифрует и дешифрует заданный текст.
5. Полученные в ходе работы навыки и умения, пригодятся в будущем для осуществления более сложных проектов, связанных с написанием крипто шифров, а также программ на языке программирования C++;

Также были сделаны выводы о том, что данная работа может пригодиться в будущем для понимания основных принципов работы криптографии, развитии людей в области информационной безопасности и программировании на языке C++, а также может послужить базой для создания более сложных проектов на основе шифра Хилла.Список используемой литературы

1. Джесси Р. Шифр Хилла.: Книга по требованию, 2013. — 74 с.: ил. ISBN: 978-5-5096-0427-0.
2. Дэвис С.Р. C++ для чайников.: Вильямс, 2018. — 400 с.: ил.ISBN: 978-5-6041393-8-7.
3. Шевцов Г.С. Линейная алгебра.: Гардарики, 1999. —360 с.: ил. ISBN: 5-8297-0013-1.
4. Habr. URL: <https://habr.com/ru/post/332714/> Дата обращения: 25.05.2021 [электронный ресурс].
5. Cyberleninka. URL: <https://cyberleninka.ru/article/n/osnovy-kriptografii-podgotovka-k-cissp/viewer> Дата обращения: 25.05.2021 [Электронный ресурс].
6. Elibrary. URL: <https://www.elibrary.ru/item.asp?id=43137648> Дата обращения: 25.05.2021 [Электронный ресурс].
7. E-olymp. URL: <https://www.e-olymp.com/ru/blogs/posts/18> Дата обращения: 25.05.2021 [Электронный ресурс].

# Приложение

## Рисунки

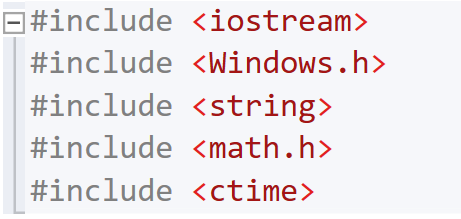


Рис. 1

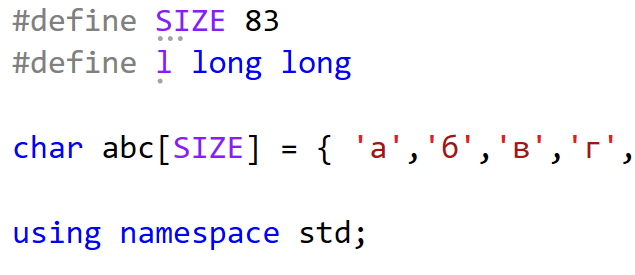


Рис. 2

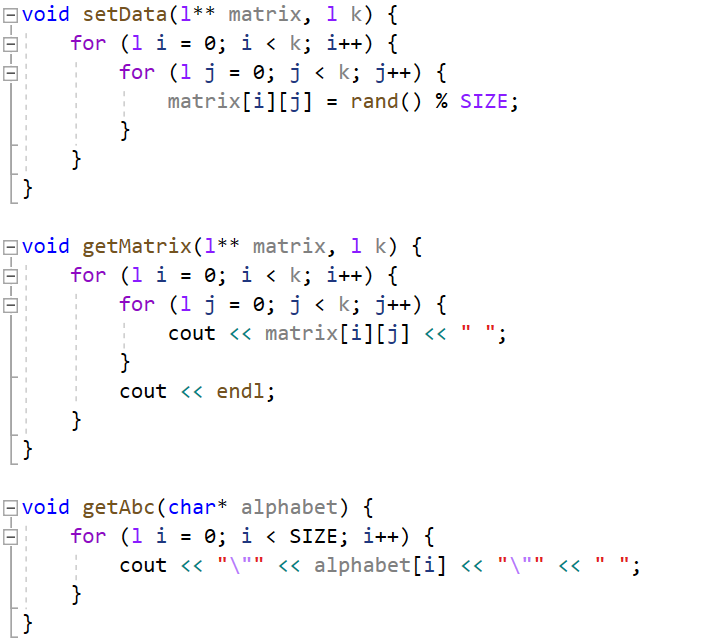


Рис. 3а

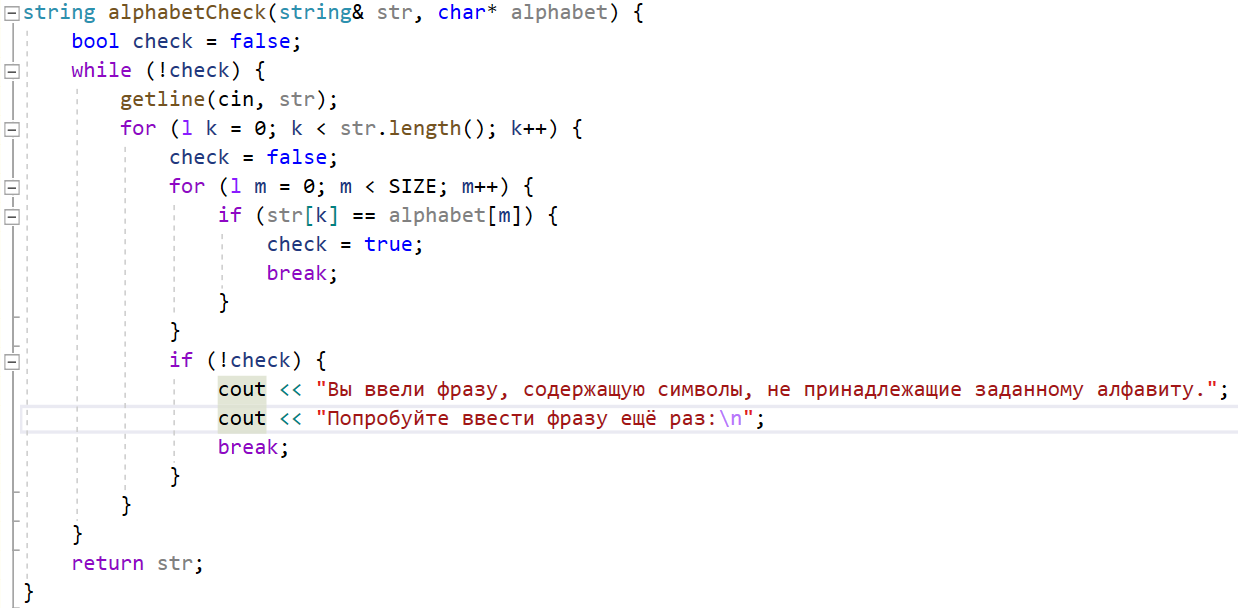


Рис. 3б

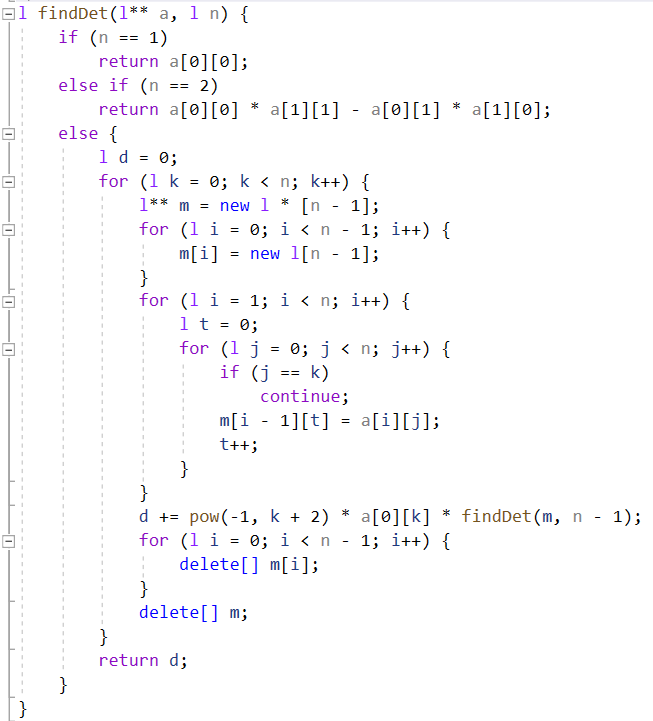


Рис. 3в

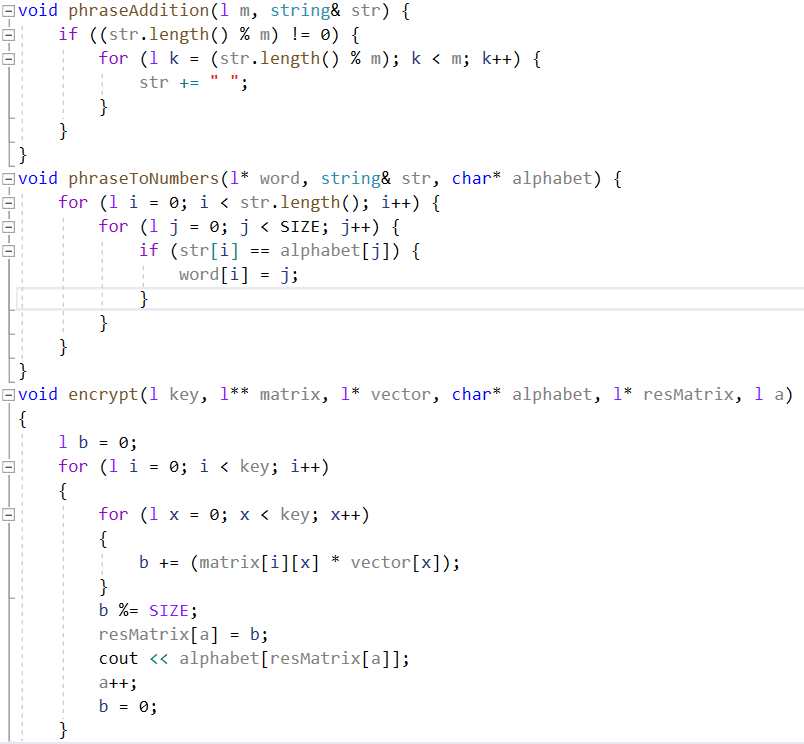


Рис. 3г

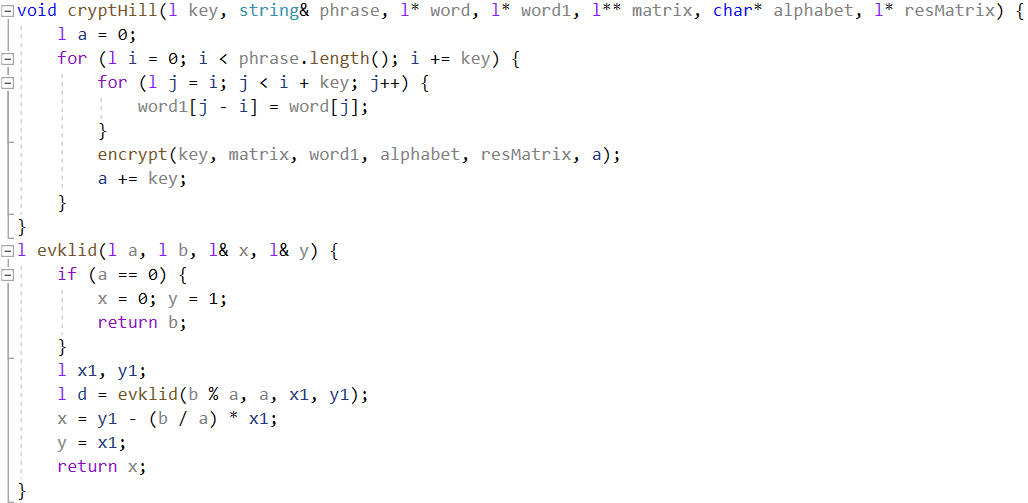


Рис. 3д

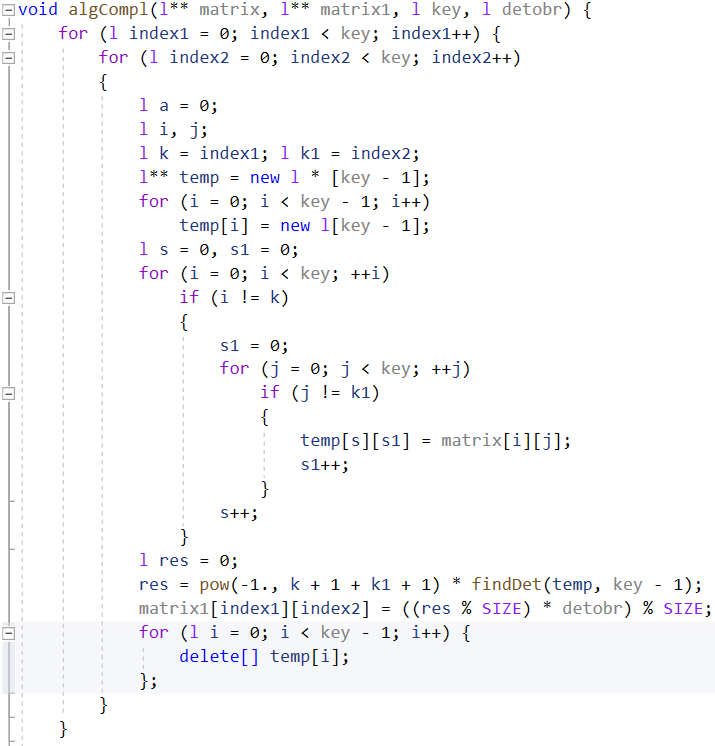


Рис. 3е

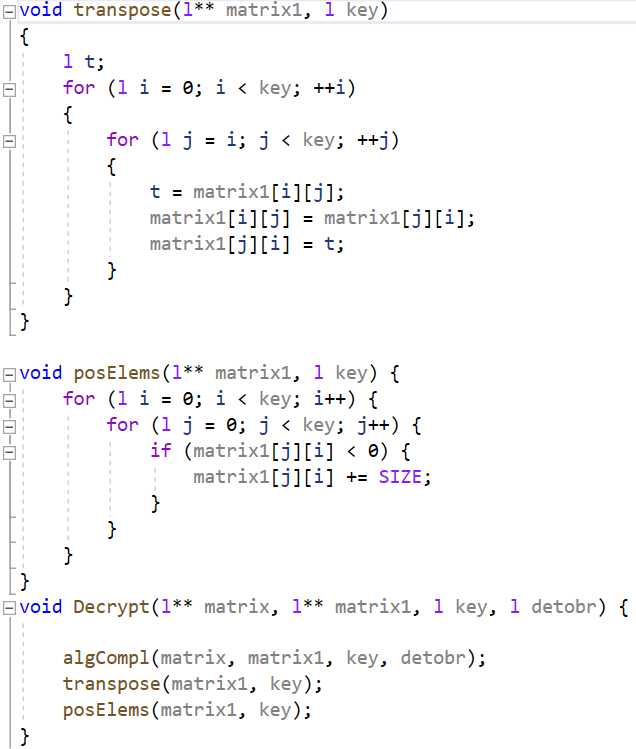


Рис. 3ж

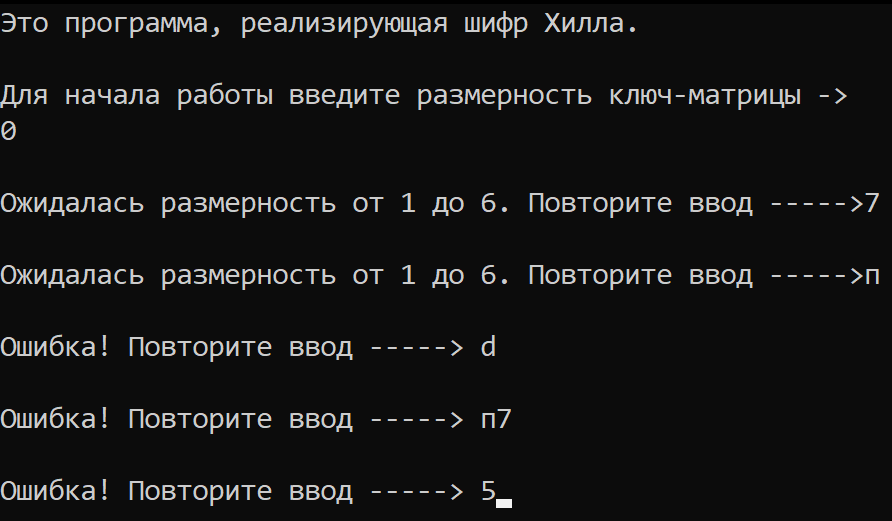


Рис. 4а

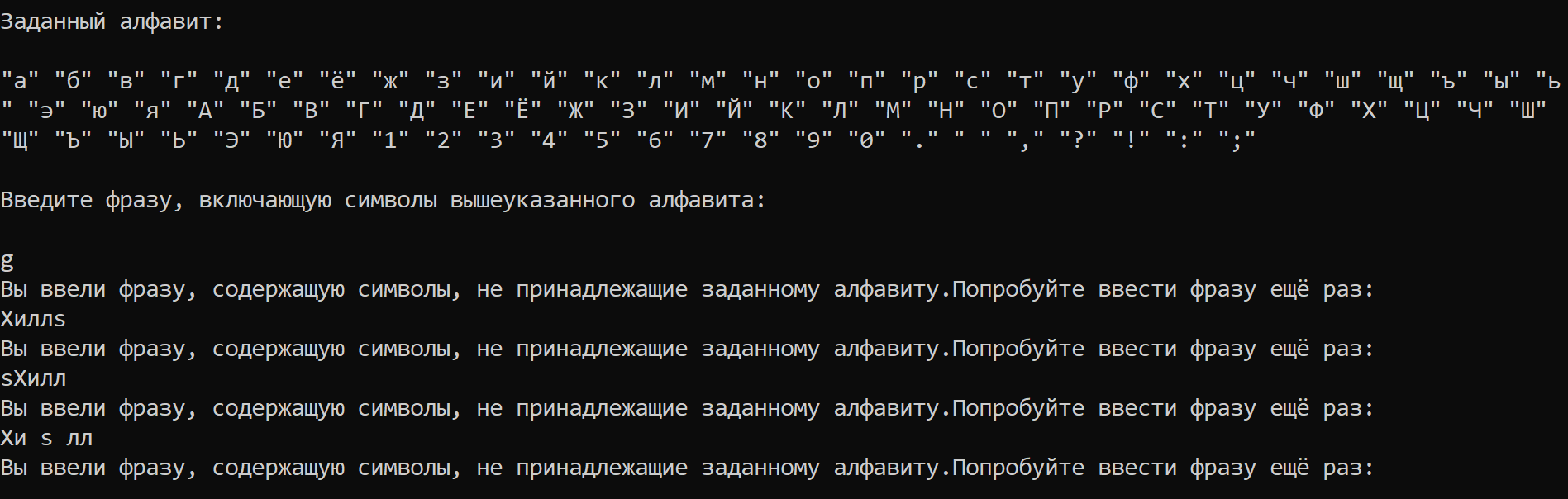


Рис. 4б

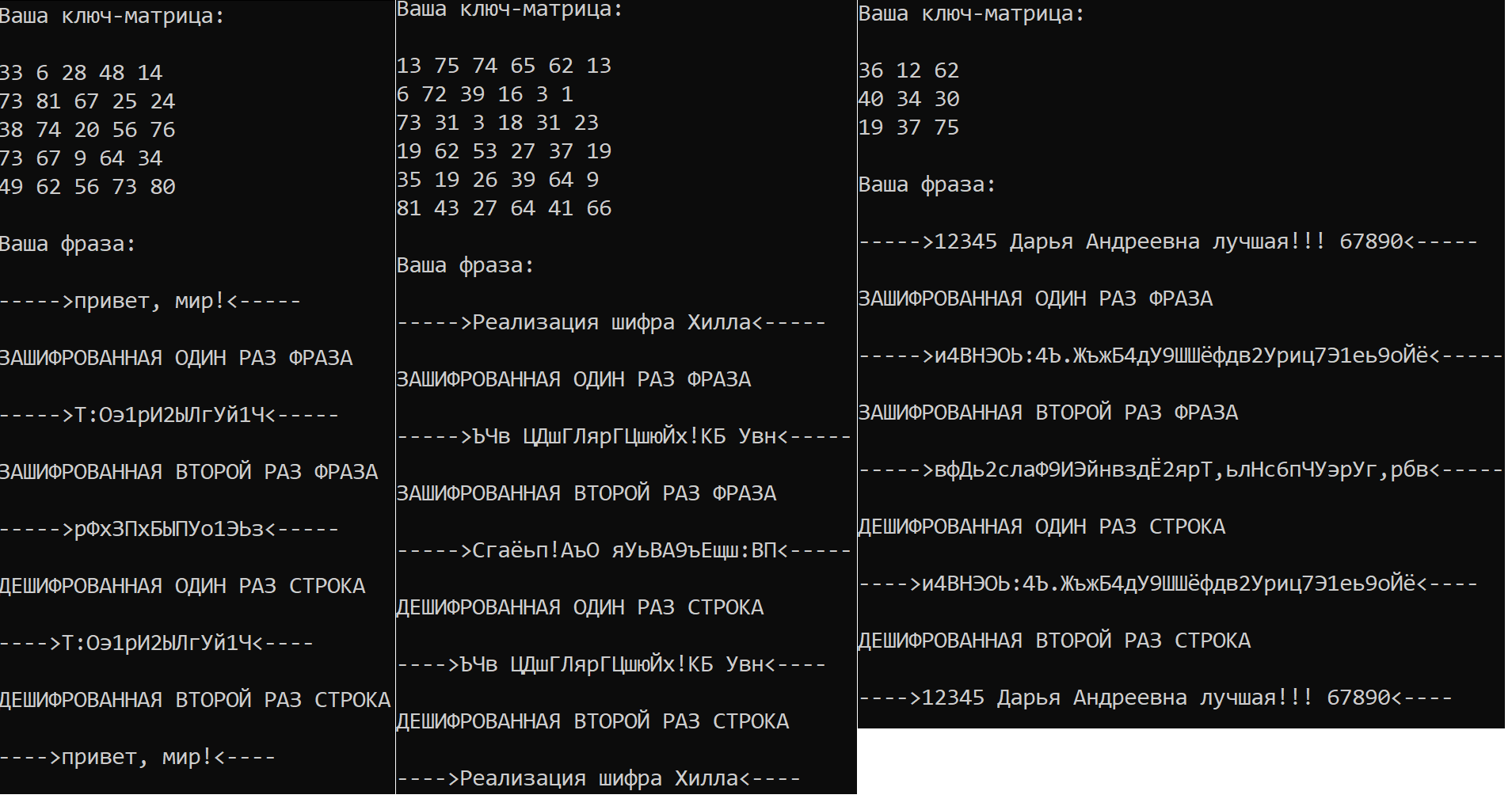


Рис. 5

## Листинг

/// Шифр Хилла

#include <iostream>

#include <Windows.h>

#include <string>

#include <math.h>

#include <ctime>

#define SIZE 83

#define l long long

char abc[SIZE] = { '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','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','1','2','3','4','5','6','7','8','9','10','.',' ',',','?','!',':',';' };

using namespace std;

void setData(l\*\* matrix, l k) {

for (l i = 0; i < k; i++) {

for (l j = 0; j < k; j++) {

matrix[i][j] = rand() % SIZE;

}

}

}

void getMatrix(l\*\* matrix, l k) {

for (l i = 0; i < k; i++) {

for (l j = 0; j < k; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void getAbc(char\* alphabet) {

for (l i = 0; i < SIZE; i++) {

cout << "\"" << alphabet[i] << "\"" << " ";

}

}

string alphabetCheck(string& str, char\* alphabet) {

bool check = false;

while (!check) {

getline(cin, str);

for (l k = 0; k < str.length(); k++) {

check = false;

for (l m = 0; m < SIZE; m++) {

if (str[k] == alphabet[m]) {

check = true;

break;

}

}

if (!check) {

cout << "Вы ввели фразу, содержащую символы, не принадлежащие заданному алфавиту.";

cout << "Попробуйте ввести фразу ещё раз:\n";

break;

}

}

}

return str;

}

l findDet(l\*\* a, l n) {

if (n == 1)

return a[0][0];

else if (n == 2)

return a[0][0] \* a[1][1] - a[0][1] \* a[1][0];

else {

l d = 0;

for (l k = 0; k < n; k++) {

l\*\* m = new l \* [n - 1];

for (l i = 0; i < n - 1; i++) {

m[i] = new l[n - 1];

}

for (l i = 1; i < n; i++) {

l t = 0;

for (l j = 0; j < n; j++) {

if (j == k)

continue;

m[i - 1][t] = a[i][j];

t++;

}

}

d += pow(-1, k + 2) \* a[0][k] \* findDet(m, n - 1);

for (l i = 0; i < n - 1; i++) {

delete[] m[i];

}

delete[] m;

}

return d;

}

}

void phraseAddition(l m, string& str) {

if ((str.length() % m) != 0) {

for (l k = (str.length() % m); k < m; k++) {

str += " ";

}

}

}

void phraseToNumbers(l\* word, string& str, char\* alphabet) {

for (l i = 0; i < str.length(); i++) {

for (l j = 0; j < SIZE; j++) {

if (str[i] == alphabet[j]) {

word[i] = j;

}

}

}

}

void encrypt(l key, l\*\* matrix, l\* vector, char\* alphabet, l\* resMatrix, l a)

{

l b = 0;

for (l i = 0; i < key; i++)

{

for (l x = 0; x < key; x++)

{

b += (matrix[i][x] \* vector[x]);

}

b %= SIZE;

resMatrix[a] = b;

cout << alphabet[resMatrix[a]];

a++;

b = 0;

}

}

void cryptHill(l key, string& phrase, l\* word, l\* word1, l\*\* matrix, char\* alphabet, l\* resMatrix) {

l a = 0;

for (l i = 0; i < phrase.length(); i += key) {

for (l j = i; j < i + key; j++) {

word1[j - i] = word[j];

}

encrypt(key, matrix, word1, alphabet, resMatrix, a);

a += key;

}

}

l evklid(l a, l b, l& x, l& y) {

if (a == 0) {

x = 0; y = 1;

return b;

}

l x1, y1;

l d = evklid(b % a, a, x1, y1);

x = y1 - (b / a) \* x1;

y = x1;

return x;

}

l findDetobr(l det, l X) {

l detobr;

if ((det < 0 && X>0) || det > 0 && X > 0) {

detobr = X;

}

else if (det > 0 && X < 0) {

detobr = SIZE + X;

}

else {

detobr = -X;

}

return detobr;

}

void algCompl(l\*\* matrix, l\*\* matrix1, l key, l detobr) {

for (l index1 = 0; index1 < key; index1++) {

for (l index2 = 0; index2 < key; index2++)

{

l a = 0;

l i, j;

l k = index1; l k1 = index2;

l\*\* temp = new l \* [key - 1];

for (i = 0; i < key - 1; i++)

temp[i] = new l[key - 1];

l s = 0, s1 = 0;

for (i = 0; i < key; ++i)

if (i != k)

{

s1 = 0;

for (j = 0; j < key; ++j)

if (j != k1)

{

temp[s][s1] = matrix[i][j];

s1++;

}

s++;

}

l res = 0;

res = pow(-1., k + 1 + k1 + 1) \* findDet(temp, key - 1);

matrix1[index1][index2] = ((res % SIZE) \* detobr) % SIZE;

for (l i = 0; i < key - 1; i++) {

delete[] temp[i];

};

}

}

}

void transpose(l\*\* matrix1, l key)

{

l t;

for (l i = 0; i < key; ++i)

{

for (l j = i; j < key; ++j)

{

t = matrix1[i][j];

matrix1[i][j] = matrix1[j][i];

matrix1[j][i] = t;

}

}

}

void posElems(l\*\* matrix1, l key) {

for (l i = 0; i < key; i++) {

for (l j = 0; j < key; j++) {

if (matrix1[j][i] < 0) {

matrix1[j][i] += SIZE;

}

}

}

}

void Decrypt(l\*\* matrix, l\*\* matrix1, l key, l detobr) {

algCompl(matrix, matrix1, key, detobr);

transpose(matrix1, key);

posElems(matrix1, key);

}

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

l key;

cout << "Это программа, реализирующая шифр Хилла.\n\nДля начала работы введите размерность ключ-матрицы -> ";

cin >> key;

while (key < 1 || key >6) {

if (cin.fail()) {

cout << "\nОшибка! Повторите ввод -----> ";

cin.clear();

cin.ignore(32767, '\n');

cin >> key;

continue;

}

if (key < 1 || key >6) {

cout << "\nОжидалась размерность от 1 до 6. Повторите ввод ----->";

cin.ignore(32767, '\n');

cin >> key;

}

}

l\*\* matrix;

matrix = new l \* [key];

for (l i = 0; i < key; i++) {

matrix[i] = new l[key];

}

l\*\* matrix1;

matrix1 = new l \* [key];

for (l i = 0; i < key; i++) {

matrix1[i] = new l[key];

}

l\*\* matrix2;

matrix2 = new l \* [key];

for (l i = 0; i < key; i++) {

matrix2[i] = new l[key];

}

setData(matrix, key);

l det = findDet(matrix, key);

while ((det == 0) || (det < INT\_MIN + 1) || (det > INT\_MAX - 1)) {

cout << "\n\nдетерминант матрицы равен 0 или превышает границы l, создам новую\n\n";

setData(matrix, key);

det = findDet(matrix, key);

}

system("cls");

cout << "Сгенерированная ключ-матрица: \n\n";

getMatrix(matrix, key);

cout << "\nЗаданный алфавит:\n\n";

getAbc(abc);

cout << "\n\nВведите фразу, включающую символы вышеуказанного алфавита:\n\n";

string phrase;

cin.ignore();

alphabetCheck(phrase, abc);

l q = phrase.length();

l z = (key - (phrase.length() % key)) % key;

phraseAddition(key, phrase);

l\* word = new l[phrase.length()];

phraseToNumbers(word, phrase, abc);

l\* word1 = new l[key];

l\* resMatrix = new l[phrase.length()];

for (l i = 0; i < phrase.length(); i++) {

resMatrix[i] = 0;

}

system("cls");

cout << "Ваша ключ-матрица:\n\n";

getMatrix(matrix, key);

cout << "\nВаша фраза:\n\n----->";

for (int i = 0; i < q; i++) {

cout << phrase[i];

}

cout << "<-----";

cout << "\n\nЗАШИФРОВАННАЯ ОДИН РАЗ ФРАЗА\n\n----->";

cryptHill(key, phrase, word, word1, matrix, abc, resMatrix);

cout << "<-----";

for (l i = 0; i < phrase.length(); i++) {

word[i] = 0;

}

for (l i = 0; i < phrase.length(); i++) {

word[i] = resMatrix[i];

}

for (l i = 0; i < phrase.length(); i++) {

resMatrix[i] = 0;

}

cout << "\n\nЗАШИФРОВАННАЯ ВТОРОЙ РАЗ ФРАЗА\n\n----->";

cryptHill(key, phrase, word, word1, matrix, abc, resMatrix);

cout << "<-----";

l x, y, X;

X = evklid(det, SIZE, x, y);

l detobr = findDetobr(det, X);

Decrypt(matrix, matrix1, key, detobr);

l\* newMatr = new l[phrase.length()];

cout << "\n\nДЕШИФРОВАННАЯ ОДИН РАЗ СТРОКА\n\n---->";

cryptHill(key, phrase, resMatrix, word1, matrix1, abc, newMatr);

cout << "<----";

Decrypt(matrix, matrix2, key, detobr);

cout << "\n\nДЕШИФРОВАННАЯ ВТОРОЙ РАЗ СТРОКА\n\n---->";

cryptHill(key, phrase, word, word1, matrix2, abc, newMatr);

for (l i = 0; i < z; i++) {

cout << "\b";

}

cout << "<----\n\n";

system("pause");

for (l i = 0; i < key; i++) {

delete[] matrix[i];

}

for (l i = 0; i < key; i++) {

delete[] matrix1[i];

}

for (l i = 0; i < key; i++) {

delete[] matrix2[i];

}

delete[] word;

delete[] word1;

delete[] resMatrix;

delete[] newMatr;

}