# **Лабораторная работа №3**

**Тема «Элементы теории информации. Информативность данных в различных кодировках»**

**Цель:** приобретение практических навыков трансформации данных и сопоставление энтропийных свойств используемых при этом алфавитов.

**Теоретические сведения**

Из энтропийных оценок (алфавитов и сообщений), полученных в ходе выполнения лабораторной работы № 2, мы выяснили, что энтропия зависит от статических характеристик самих алфавитов и сообщений (вспомним энтропию по Шеннону и по Хартли).

Энтропия максимальна при равномерном появлении букв на любом месте сообщения. Для характеристики источника сообщений с различным алфавитом представляет интерес сравнение фактической энтропии источника с максимально возможной. В этом смысле введено понятие избыточности источника сообщений, или избыточности алфавита.

**Избыточностью алфавита** называют уменьшение информационной нагрузки на один символ вследствие разной вероятности и взаимозависимости появления его символов в сообщениях.

В наиболее общем виде избыточность алфавита R можно оценить отношением энтропии по Хартли и по Шеннону; при этом первая рассчитывается по выражению (2.2), вторая – по формуле (2.1):



При выполнении предыдущей работы мы убедились, что формально одно и то же сообщение, но представленное на основе алфавита русского (белорусского, английского или иного) языка – с одной стороны, и представленное в кодах ASCII – с другой, будут характеризоваться различным количеством содержащейся в них информации. Эта дополнительная избыточность обусловлена переносом сообщения из одной среды в другую или, иначе говоря, кодированием символов исходного алфавита.

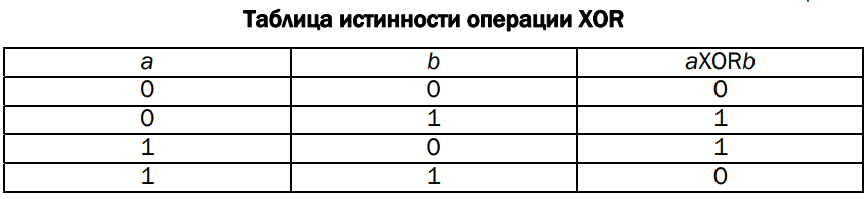
Утверждение восьмибитных кодировок (ASCII) как стандарта принесло некоторые проблемы. К этому моменту уже существовала определенная инфраструктура, использующая семибитные кодировки. Известны проблемы с «обрезанием восьмого бита» в системе электронной почты. Утверждение восьмибитного символа дало 256 различных значений, что позволило уместить в одной кодовой таблице и общепринятые символы (цифры, знаки препинания, латиницу), и символы кириллицы.

Уже созданное к тому времени и работающее программное обеспечение зачастую было приспособлено для семибитных кодировок, что приводило, например, к тому, что почтовый сервер при передаче письма обнулял старшие биты в каждом байте сообщения.

Одним из решений проблемы стала кодировка (а точнее – алгоритм) base64. В PGP алгоритм base64 используется для кодирования бинарных данных. Кодирование base64 разработано для представления произвольных последовательностей октетов в форме, позволяющей использовать строчные и прописные буквы. Используется 65-символьное подмножество набора символов US-ASCII, обеспечивающее представление одним печатным символом 6 битов данных (дополнительный 65-й символ используется для обозначения функции специальной обработки).

Kаждые 6 битов буфера, начиная с самых старших, используются как индексы строки «ABCDEFGHIJKLMNOPQRSTUV WXYZabcdefghijklmnopqrstuvwxyz0123456789+/», и ее символы, на которые указывают индексы, помещаются в выходную строку.

**Если кодируются только один или два байта, в результате получаются только первые два или три символа строки, а выходная строка дополняется двумя или одним символами «=». Это предотвращает добавление дополнительных битов к восстановленным данным.**

****

Процесс повторяется над оставшимися входными данными. Такая обработка выполняется в тех случаях, когда последняя группа входных данных содержит меньше 24 битов. Кодируемое значение всегда завершается полным квантом кодирования.

Если на входе доступно менее 24 битов, входная группа дополняется (справа) нулями до формирования целого числа 6-битных групп. Заполнение в конце данных осуществляется как раз с использованием символа «=». Поскольку входная информация base64 всегда включает целое число октетов, возможны лишь перечисленные ниже случаи:

* размер финального блока кодирования на входе кратен 24 битам, кодированный результат будет содержать целое число 4-символьных групп без заполнения символами «=»;
* размер финального блока кодирования на входе составляет 8 битов, выходной блок будет представлять 2 символа, дополненные последовательностью из двух символов заполнения «==»;
* размер финального блока кодирования на входе составляет 16 битов, выходной блок будет представлять 3 символа, дополненные символом заполнения «=».

Если а и b имеют длину более 1 бита, к примеру 1 байт, то рассматриваемая операция над ними выполняется побитово. Указанным байтам могут соответствовать символы в определенной кодировке. Положим, символу «М» (hex4d) соответствует 8-битный код 01001101 (см. табл. 3.2), а символу «а» (hex61) соответствует код 01100001, тогда операция сложения по модулю 2 этих двух бинарных кодов дает 00101100 ((hex2с), или символ «,».

**Практическое задание**

Создать собственное приложение (приветствуется!) или воспользоваться Base64-онлайн-кодировщиком, с помощью которого конвертировать произвольный документ (а) на латинице (можно использовать документ из лабораторной работы № 1) в документ (б) формата base64.

С помощью приложения, созданного в лабораторной работе № 1, получить распределение частотных свойств алфавитов по документам (а) и (б). Вычислить энтропию Хартли и Шеннона, а также избыточность алфавитов. Объяснить полученный результат.

Написать функцию, которая принимает в качестве аргументов два буфера (а и b) одинакового размера и возвращает XOR (собственная фамилия (а) и имя (b); при разной длине меньшую дополнить нулями). Входные аргументы представлять: 1) в кодах ASCII; 2) в кодах base64. Что будет результатом операции аXORbXORb?

При написании не использовать стандартные функции языка программирования. Итоговые данные сравнить с результатами использования стандартных функций языка программирования (если они есть).

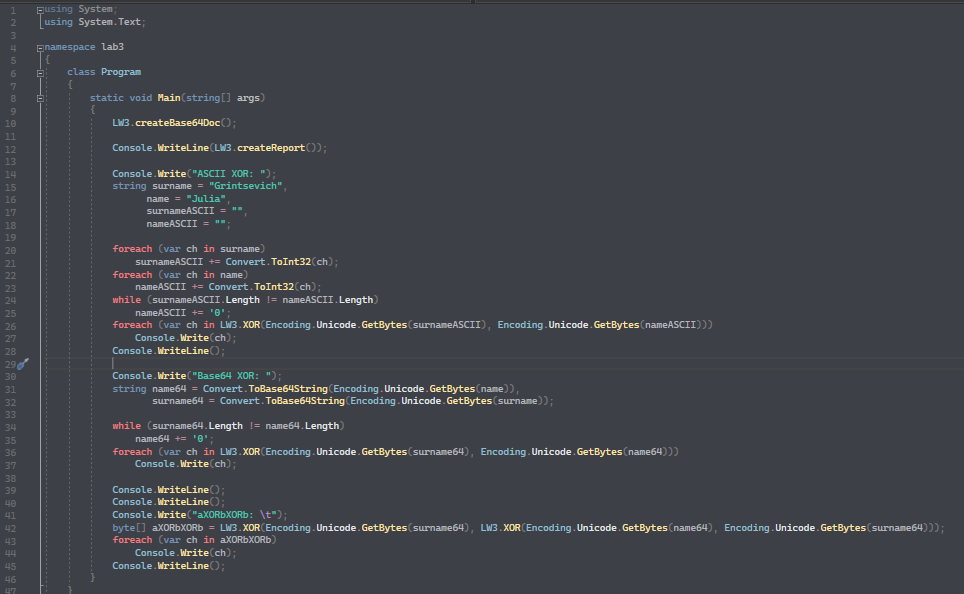
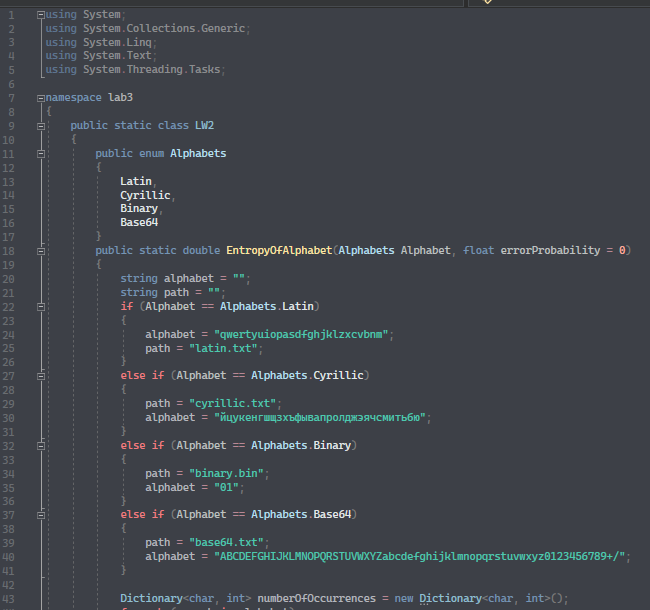
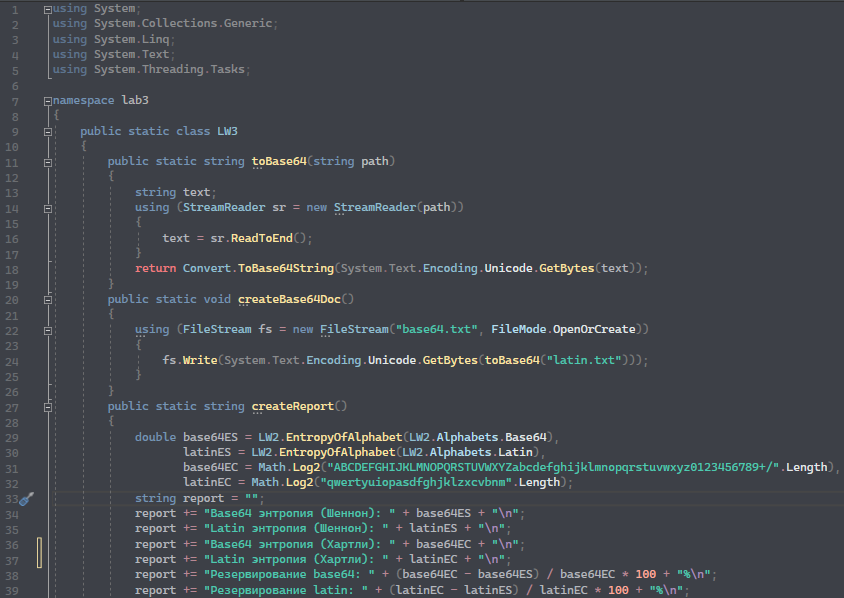
****

Рис. *1* – Код файла Program.cs

****

****

Рис. *2* – Код файла LW2.cs

****

****

Рис. *2* – Код файла LW3.cs

**Ответы на вопросы:**

1. **Что такое base64?**

**Base64** - это группа схожих binary-to-text encoding схем, которые представляют двоичные данные в ASCII-формате методом перевода в radix-64 представление. Термин Base64 происходит от a specific MIME content transfer encoding.

Кодирование Base64 широко используется в случаях, когда требуется перекодировать двоичные данные для передачи по каналу приспособленному для передачи текстовых данных.

. Используется 65-символьное подмножество набора символов US-ASCII, обеспечивающее представление одним печатным символом 6 битов данных (дополнительный 65-й символ используется для обозначения функции специальной обработки).

Избыточностью алфавита называется уменьшение информационной нагрузки на один символ вследствие неравновероятности и взаимозависимости появления его символов.

Информационная избыточность характеризует относительную нагруженность алфавита

1. **Как проверить, была ли определенная строка символов закодирована в base64?**

 Мы можем использовать следующее регулярное выражение, чтобы проверить, закодирована ли строка в base64 или нет:

**^([A-Za-z0-9+/]{4})\*([A-Za-z0-9+/]{3}=|[A-Za-z0-9+/]{2}==)?$**

В кодировке base64 набор символов - [A-Z, a-z, 0-9 и + /]. Если остальная длина меньше 4, строка заполняется символами '='.

^([A-Za-z0-9+/]{4})\* означает, что строка начинается с 0 или более групп base64.

([A-Za-z0-9+/]{4}|[A-Za-z0-9+/]{3}=|[A-Za-z0-9+/]{2}==)$ означает, что строка заканчивается одной из трех форм: [A-Za-z0-9+/]{4}, [A-Za-z0-9+/]{3}= или [A-Za-z0-9+/]{2}==.

1. **Как с помощью base64 проверить подлинность вводимых данных в форму пароля и логина?**
2. **Охарактеризовать энтропийные свойства алфавитов в проанализированных форматах данных.**

*Энтропия алфавита - это количество информации, приходящееся на один символ.* Другими словами, это информационная нагрузка, которую несет один символ алфавита.

1. **Объяснить результат операции аXORbXORb. Где может найти применение такая операция?**

Эта операция называется также cложением по модулю 2, логическим сложением, исключающим «ИЛИ», строгой дизъюнкций, поразрядным дополнением.

1. **Как будут выглядеть строки: efd8b295a633908a3c0828b2 faea8766 4d72cde3aaa0 после их конвертации в base64?**
2. **Результатом операции аXORb (а – каждый байт строки, b – некоторая неизменная величина) будет строка:**

**1f180d1e1f04051c404c0f19**

**1f180308050d024c030a4c18**

**04094c1f18030009024c1c00**

**Найти значение b.**

**Выводы:** приобрела практические навыки трансформации данных и сопоставление энтропийных свойств используемых при этом алфавитов.

# **Лабораторная работа №4**

**Тема «Избыточное кодирование данных в информационных системах. Код Хемминга»**

**Цель:** приобретение практических навыков кодирования/декодирования двоичных данных при использовании кода Хемминга.

**Ответы на вопросы:**

**Вывод:** приобрела практические навыки кодирования/декодирования двоичных данных при использовании кода Хемминга.

# **Лабораторная работа №5**

**Тема «Избыточное кодирование данных в информационных системах»**

**Цель:** приобретение практических навыков кодирования/декодирования двоичных данных при использовании итеративных кодов.

**Теоретические сведения**

**Ответы на вопросы:**

**Выводы:** приобрела практические навыки кодирования/декодирования двоичных данных при использовании итеративных кодов.

# **Лабораторная работа №6**

**Тема «Избыточное кодирование данных в информационных системах. Циклические коды»**

**Цель:** приобретение практических навыков кодирования/декодирования двоичных данных при использовании циклических кодов (ЦК).

**Ответы на вопросы:**

**Выводы:** приобрела практические навыки кодирования/декодирования двоичных данных при использовании циклических кодов (ЦК).

# **Лабораторная работа №7**

**Тема «Перемежение/деперемежение данных в информационно-вычислительных системах»**

**Цель:** приобретение практических навыков использования методов перемежения/деперемежения двоичных данных в информационных системах.

**Теоретические сведения**

**Ответы на вопросы:**

**Вывод:** приобрела практические навыки использования методов перемежения/деперемежения двоичных данных в информационных системах.

# **Лабораторная работа №8**

**Тема «Сжатие/распаковка данных методом Барроуза - Уилера»**

**Цель:** приобретение практических навыков использования метода Барроуза − Уилера для сжатия/распаковки данных.

**Теоретические сведения**

**Ответы на вопросы:**

**Вывод:** приобрела практические навыки использования метода Барроуза − Уилера для сжатия/распаковки данных.

# **Лабораторная работа №9**

**Тема «Сжатие/распаковка данных на основе статических методов»**

**Цель:** приобретение практических навыков использования статистических методов Шеннона − Фано и Хаффмана (ShannonFano and Huffman coding) для сжатия/распаковки данных.

**Ответить на вопросы:**

**Вывод**: приобрела практические навыкови использования статистических методов Шеннона − Фано и Хаффмана (ShannonFano and Huffman coding) для сжатия/распаковки данных

# **Лабораторная работа №10**

**Тема «Сжатие/распаковка данных методов Лемпеля-Зива»**

**Цель:** приобретение практических навыков использования метод Лемпеля − Зива (Lempel-Ziv) для сжатия/распаковки данных.

**Ответить на вопросы:**

**Вывод:** приобрела практические навыки использования метод Лемпеля − Зива (Lempel-Ziv) для сжатия/распаковки данных.

# **Лабораторная работа №11**

**Тема «Сжатие/распаковка данных арифметическим методом»**

**Цель:** приобретение практических навыков использования арифметических методов сжатия/распаковки данных.

**Теоретические сведения**

**Вывод:** приобрела практические навыки использования арифметических методов сжатия/распаковки данных.