Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Элементы теории информации. ИНФОРМАТИВНОСТЬ ДАННЫХ В РАЗЛИЧНЫХ КОДИРОВКАХ**

Студент: Скалкович Станислав

ФИТ 3 курс 2 группа

Преподаватель: Нистюк О.А.

Минск 2024

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

Задачи:

1. Закрепить теоретические знания по взаимной конвертации данных, представленных в кодах ASCII и base64.
2. Разработать приложение для конвертации произвольного документа в формат base64 и обратно.
3. Исследовать энтропийные характеристики используемых в конвертерах алфавитов.
4. Изучить особенности практической реализации операции XOR над данными, представленными в разных форматах.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

# **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).

Процесс кодирования представляет группу из 24 последовательных битов в форме строки из 4 символов. Обработка выполняется слева направо, а 24-битная исходная группа образуется конкатенацией трех 8-битных групп (байтов). Данные 24 бита после этого трактуются как 4 сцепленных группы по 6 битов, каждая из которых транслируется в один символ алфавита base64.

Каждая 6-битная группа используется в качестве индекса массива из 64 печатных символов. Символы алфавита, соответствующие индексу, помещаются в выходную строку [6]. Кодирование base64 с безопасным алфавитом используется для представления URL и имен файлов.

В табл. 3.1 перечислен алфавит, используемый для base64- кодировки. Значения представлены в различных системах счисления: десятичной (10), двоичной (2), восьмеричной (8) и шестнадцатеричной (16 или hex).

Еще раз обратимся к процессу кодировки. Как было выше установлено, каждые 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 «=».

При изучении раздела курса, касающегося криптографического преобразования данных, мы вернемся к вопросу о расширения области применения base64-формата. Сейчас же ограничимся рассмотрением особенностей дальнейшего преобразования данных этого формата с использованием операции XOR (вспомним, что эта операция называется также cложением по модулю 2, логическим сложением, исключающим «ИЛИ», строгой дизъюнкций, поразрядным дополнением). Вспомним таблицу истинности над двумя переменными: а и b (табл. 3.3).

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 «,».

# **Практические задания**

1. 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.

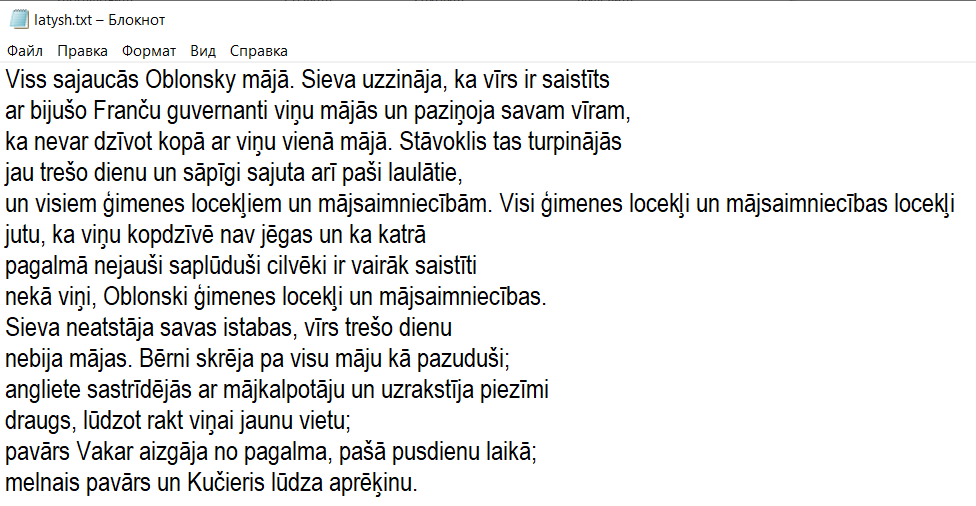


Рисунок 1 — Документ а на латышском языке

В качестве содержимого исходного файла (документа а) взят текст на латышском языке латинскими буквами.

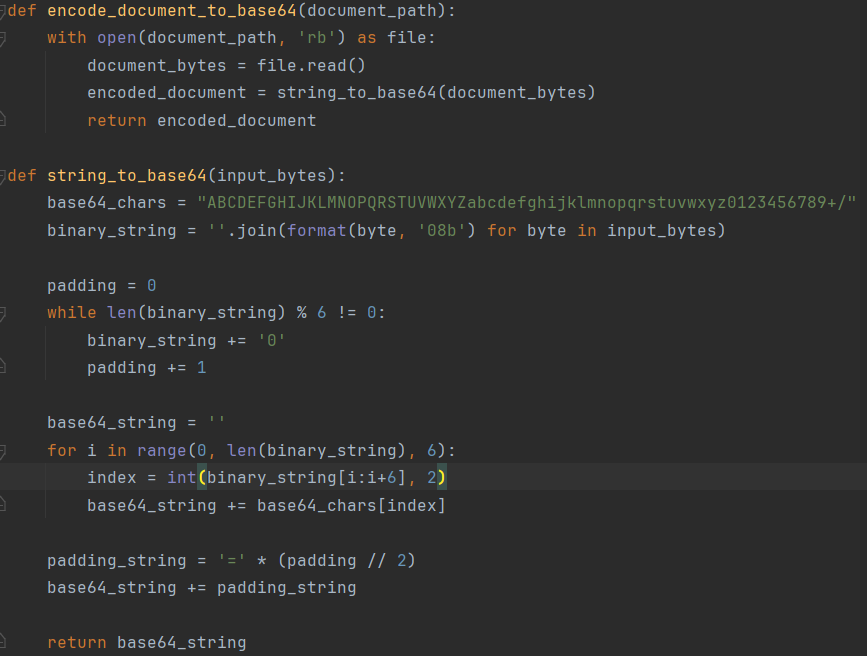
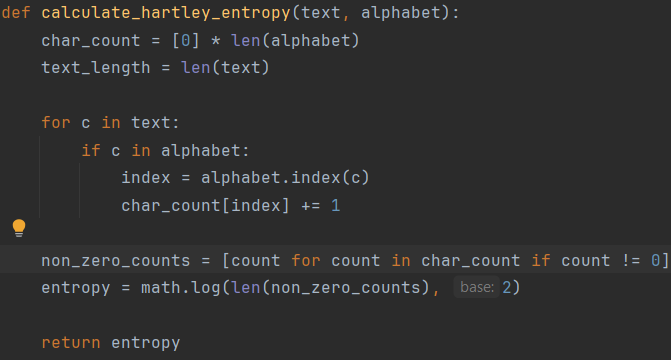


Рисунок 2 —Функции перевода латиницы в формат base64



Рисунок 3 — Документ б, результат первода латышского теста в кодировку base64

2. 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.



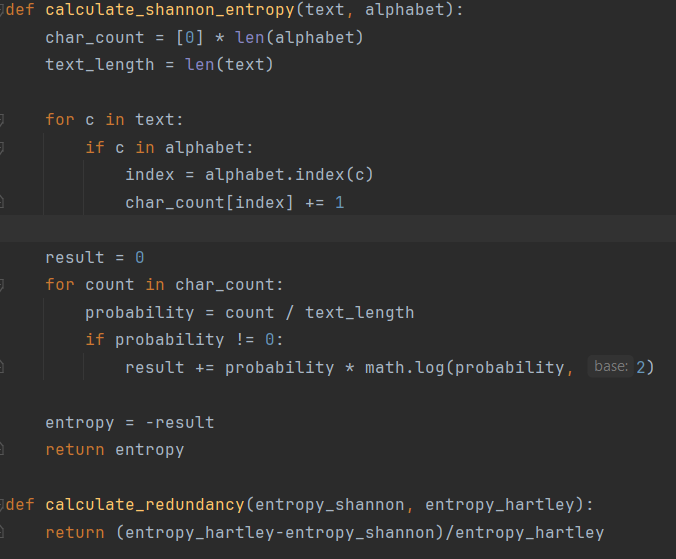


Рисунок 4 — Функции для расчета энтропии по Хартли, по Шеннону, а также избыточности алфавитов

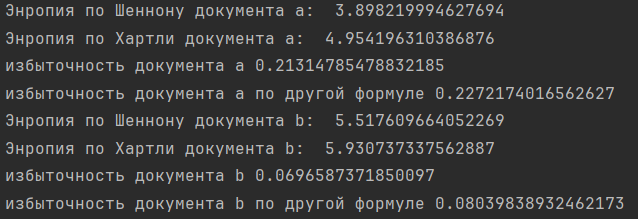


Рисунок 5 — Результат вызова функций расчета энтропии по Хартли, по Шеннону, а также избыточности алфавитов для документа а и документа b

**Вывод:** Исходя из этих результатов, можно сделать вывод, что текст на латышском языке (документ "a") имеет более низкую энтропию и более высокую избыточность по сравнению с текстом, закодированным в формате Base64 (документ "b"). Текст на латышском языке обычно содержит меньше разнообразия символов и требует больше информации для его представления. С другой стороны, текст, закодированный в формате Base64, имеет более высокую энтропию и меньшую избыточность, что связано с использованием широкого набора символов в кодировке Base64, потому что каждый символ Base64 содержит более высокую информационную плотность по сравнению с символами в обычном тексте.

3. 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

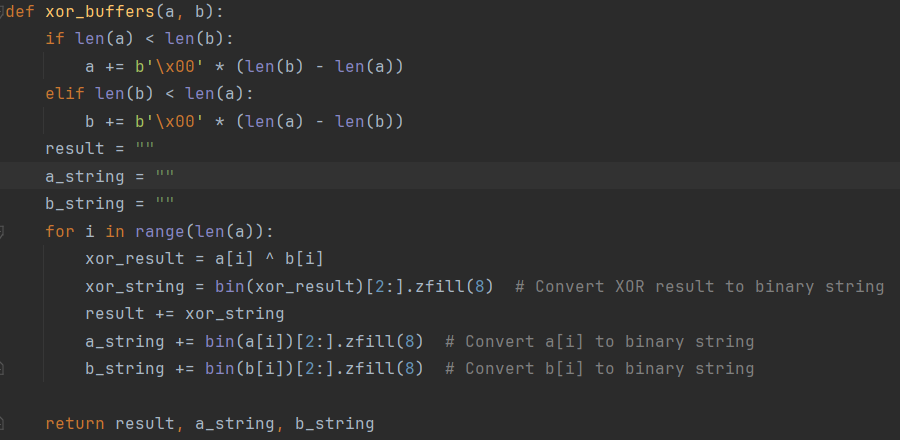


Рисунок 6 — Функция XOR-преобразования двух строк

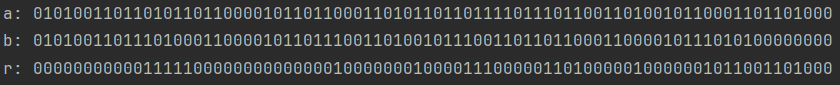


Рисунок 7 — Результат XOR-преобразования имени и фамилии в кодах ASCII

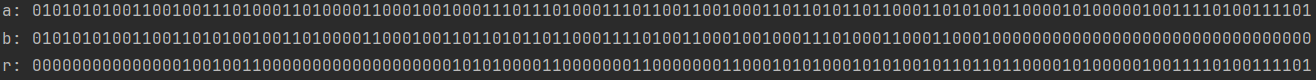


Рисунок 8 — Результат XOR-преобразования имени и фамилии в кодах base64

**Вывод**

В ходе выполнения лабораторной работы были изучены основы теории информации и приобретены практические навыки в области трансформации данных и анализа их энтропийных свойств. Было разработано приложение, позволяющее рассчитывать энтропию по Шеннону и Хартли, а также избыточность информации для алфавитов ASCII и Base64.

Также в результате выполнения лабораторной работы было разработано приложение, которое позволяет вычислять энтропию по Шеннону и Хартли, а также избыточность информации для алфавитов ASCII и Base64. Из проведенного анализа можно сделать следующие выводы:

* Когда энтропия низкая и избыточность высокая, это указывает на то, что в документе содержится значительное количество повторяющихся символов. Это делает документ менее информативным, так как большая часть информации может быть извлечена из уже прочитанного контента.
* Когда энтропия высокая и избыточность низкая, это указывает на наличие большого количества уникальных символов в документе. Такой документ обычно содержит более разнообразную информацию и может быть более информативным для анализа.

Таким образом, анализ энтропии и избыточности информации помогает оценить степень информативности документа и может быть полезным для принятия решений о его обработке или сжатии.