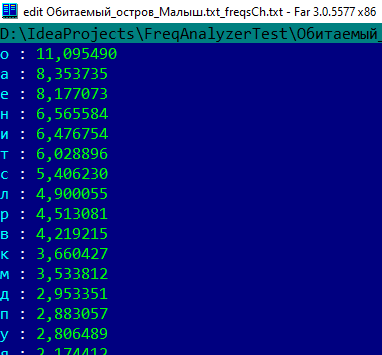
# Алгоритм дешифровки по символам (монограммам)

## Таблицы частотностей

Строятся 2 таблицы частотностей: стандартная для русского языка и таблица для шифрованного текста (текстовая таблица). Таблицы содержат соответствие: символ и частотность его появления в тексте. Обе таблицы упорядочены по убыванию частотности.

### Стандартная таблица частотностей

Стандартная таблица загружается из файла. Файл имеет вид:   


### Таблица частотностей шифрованного текста

Алгоритм построения текстовой таблицы:

1. Переводим шифрованный текст в нижний регистр.

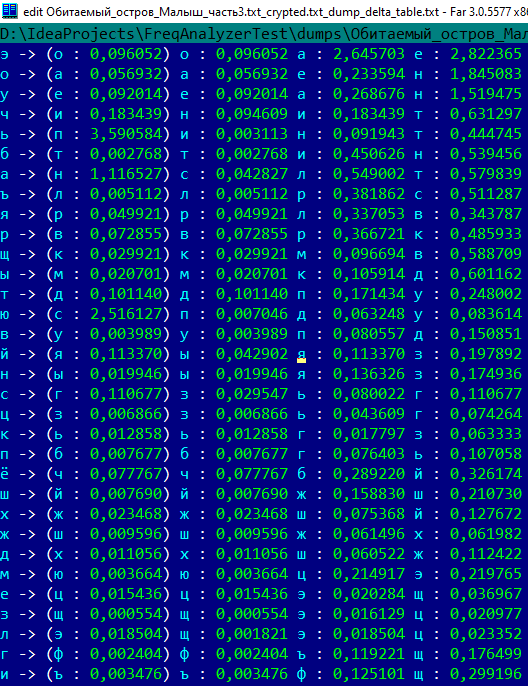
2. Составляем алфавит букв нижнего регистра русского языка.

3. Рассчитываем Fch для символов текста. Последовательно просматриваем каждый символ текста. Для тех символов, которые входят в алфавит подсчитываем сколько раз встретился каждый символ (Nch) и общее количество рассмотренных символов (N). Символы текста не входящие в алфавит не учитываются. Для каждого символа рассчитываем частотность в процентах по формуле:

Fch = (Nch/N) \* 100%

## Таблица разностей частот

**Строим таблицу разностей и частот:**



Ключом таблицы является символ шифрованного текста. В столбцах таблицы содержится разность частотностей от ключа до каждого символа из стандартной таблицы частотностей.

ΔF = abs(Fchstd- Fchtxt)

Строки таблицы упорядочены по уменьшению частотности ключа - Fchtxt. Внутри каждой строки стандартные символы упорядочены по увеличению ΔF.

## Алгоритм сопоставления символов

**Делаем сопоставление chstd и chtxt**. Каждому chtxt должен соответствовать один и только один chstd. На рисунке выше сопоставление показаны скобками. Например символу ‘э’ соответствует ‘о’.

Наилучшим соответствием chstd и chtxt  считается такое соответствие, для которого ΔF минимально.

Для этого просматриваем строки таблицы сверху вниз и столбцы таблицы слева направо. На каждом проходе по конкретному столбцу находим минимум ΔF для строки. Найденное соответствие ключа строки chtxt символу chstd из столбца считаем наилучшим и окончательным.

Например, рассматриваем 1-ю строку: chtxt = ‘э’, проходим 1-й столбец. Символу ‘э’ подходит символ ‘о’ из той же строки и больше символ ‘о’ ни в одной строке в первом столбце не встречается. Значит соответствие ‘э’ – ‘о’ наилучшее. Отмечаем его как зафиксированное.

Аналогично со всеми строками до строки 16 символ ‘й’. В строке 16 в 1-ом столбце соответствует символ ‘ы’ с частотностью 0.042902. При просмотре столбца вниз выясняется, что символ ‘ы’ лучше всего подходит для символа ‘н’, строка 17 частотность 0.019946. Значит соответствие ‘н’ – ‘ы’ наилучшее. Отмечаем его как зафиксированное.

После прохода 1 столбца без соответствия остались символы ‘й’, ‘с’. Для их наилучших соответствий, аналогично просматриваем второй столбец. Для символа ‘й’ лучше всего подходит символ ‘я’. Отмечаем его как зафиксированный. Для буквы ‘с’ во втором столбце соответствий нет. Мог бы подойти ‘ь’, но его частотность не минимальная и кроме того ‘ь’ уже зафиксирован после прохода по 1-му столбцу.

Переходим к 3-му столбцу, без соответствия остался только символ ‘с’, ему однозначно соответствует символ ‘г’ так как он еще не зафиксирован. Отмечаем соответствие ‘с’-’г’ как зафиксированное. Все строки стали зафиксированы, т.е. мы нашли наилучшее соответствие для всех ключей chtxt.

Сопоставление построено.

## Алгоритм замены символов

**Производим замену символов в шифрованном тексте по таблице сопоставлений**. Алгоритм замены: для каждого chtxt заменяем на chstd верхнего регистра (чтобы избежать неправильных замен). После того как все замены сделаны, переводим текст в нижний регистр.

# Алгоритм дешифровки по биграммам

Для биграмм используется тот же алгоритм но с отличиями:

Биграммой называется последовательность из двух символов каждый из которых входит в алфавит.

В алгоритме выше каждая биграмма рассматривается как символ. Каждый элемент таблицы это не символ а биграмма.

## Алгоритм выделения биграмм из текста

Текст разбивается на слова и строится массив слов. Словом называется любая последовательность символов входящих в алфавит, все остальные символы в тексте считаются разделителями и не учитываются. Для каждого слова сначала выделяем биграммы без смещения от начала слова, затем выделяем биграммы со смещением на 1 символ. Все остальные действия аналогичны для монограмм.

# Общий алгоритм программы дешифровки

Сначала находим наилучшее сопоставление для биграмм и берем из них первые 7 как наиболее часто встречающиеся. Переводим выбранные биграммные сопоставления, в сопоставления символов и рассматриваем их как априорные сопоставления. Для остальных символов строим сопоставление по алгоритму указанному выше. По построенной таблице сопоставлений дешифруем текст.

# Результаты работы программы

## Исходные данные

Исходным текстом являлся текст А. и Б. Стругацких “Обитаемый остров” и “Малыш”. По полному тексту строилась стандартная таблица частот, далее был взят текст - начало обитаемого острова, зашифрован шифром Цезаря и расшифрован.

## Результат расшифровки

Результат работы программы выводится в файл output.txt. Сначала выводится шифрованный текст (шифром Цезаря с ключом 15). Далее показаны первые 7 наивероятнейших биграмм и перевод их в априорные монограммы. Внизу показан расшифрованный текст.

