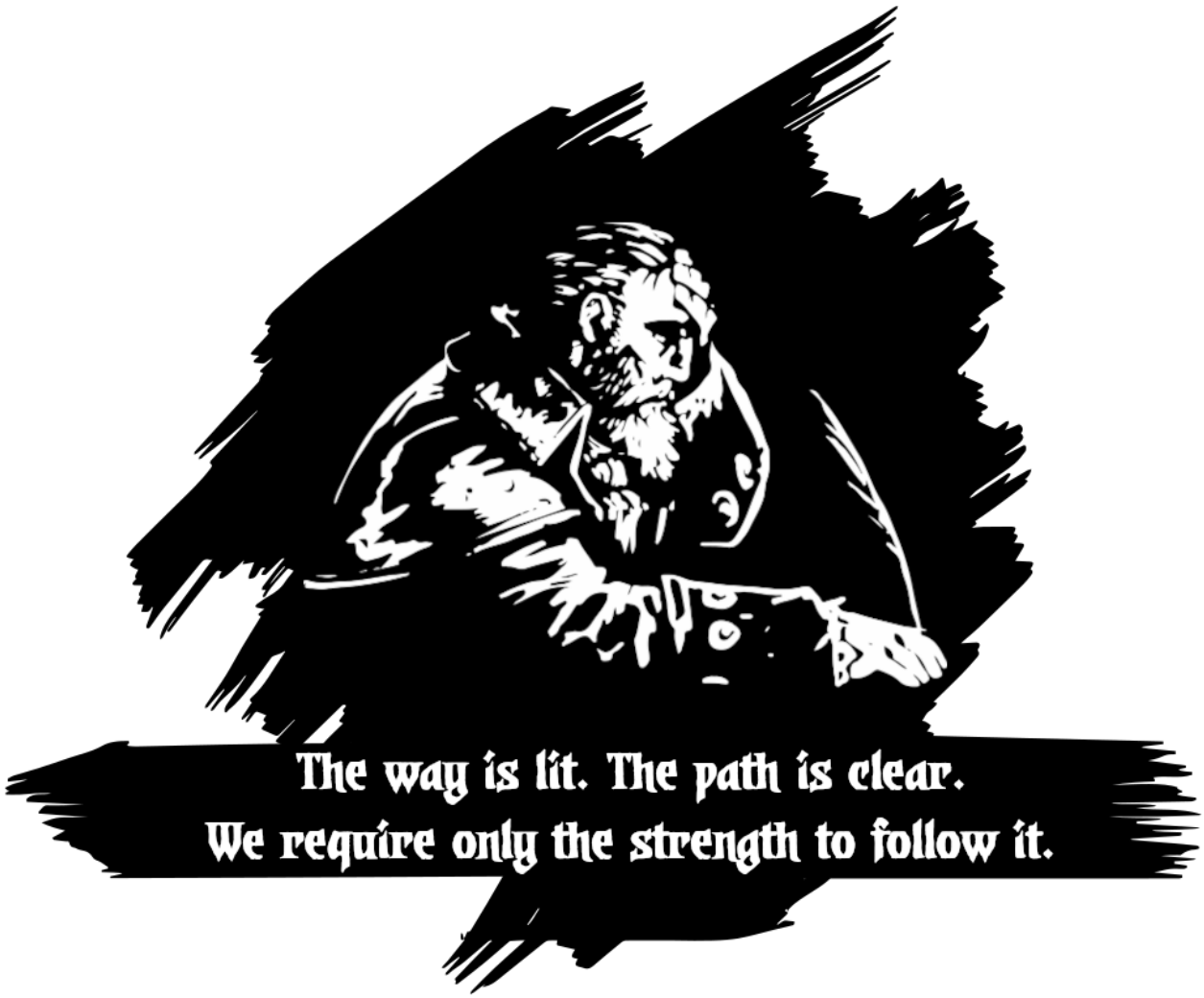
Exercises — Frequency Analysis

version **#7be580532266ed398481e31366afcc24b1950c2a**



Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.assistants.epita.fr

File Tree

```
freq_analysis/
└── freq_analysis.c  (to submit)
```

Authorized functions : You are only allowed to use the following functions

- printf(3)

Authorized headers : You are only allowed to use the functions defined in the following headers

- err.h
- errno.h
- assert.h
- stddef.h

Compilation : Your code must compile with the following flags

- -std=c99 -pedantic -Werror -Wall -Wextra -Wvla

Main function : None

1 Presentation

The monoalphabetic substitution cipher is a method that replaces each distinct character of the original message with a different character. For example, all the letters `A` of the message may be substituted with the letter `X`. Once all characters in the text are replaced, only the person knowing all the associations between symbols will be able to perform the reverse operation to decipher the message.

Fortunately, this method is no longer used because it is possible to recover the original message without knowing the associations only by observing the occurrence frequency of different symbols of the encrypted message.

In French, for example, the `E` letter appears much more frequently than others; so it makes sense that in a long enough encrypted message, the most used symbol corresponds to `E`.

There are tables for each language listing the occurrence frequencies of each letter or each set of letters (`ES` in French or `TH` in English are for example very used *bigrams*) that can easily break such substitution methods.

2 Goal

In this exercise, you have to write a function taking two arguments:

1. A text ciphered by a monoalphabetic substitution method.
2. A table listing the characters of the original message, from the most common to the least common.

Prototype:

```c
void freq_analysis(const char text[], const char table[]);
```

You must display on the standard output the associations `ciphered letter` $\iff$ `original letter` (one per line, in **alphabetical order**). For example, if a `X` in `text[]` matches a `E` in the original message, you must display "X E" followed by a newline.

For your convenience:

- `text[]` and `table[]` only contain characters in the '[A-Z]' range;
- all inputs are valid;
- each character in `table` will be unique.
- strlen(`table`) = number of unique characters in `text`.

The `freq_analysis.c` file must not include a `main` function. You may use a `main.c` file for your tests.

3 Examples

```
42sh$ ./freq_analysis 'AAB' 'XY' | cat -e
A X$
B Y$
42sh$ ./freq_analysis 'ABBCCCDDDDD' 'ABCD' | cat -e
A D$
B C$
C B$
D A$
42sh$ ./freq_analysis 'FXOWFFOWOFF' 'ABCD' | cat -e
F A$
O B$
W C$
X D$
```

4 Detail

If two letters have the same number of occurrences in `text`, you must associate them with their corresponding characters in `table` in the same relative order : the first to appear in `text` will be associated with the first to appear in `table`.

4.1 Example

```
42sh$ ./freq_analysis 'CAABB' 'XYZ' | cat -e
A X$
B Y$
C Z$
```

The way is lit. The path is clear. We require only the strength to follow it.