Programming Languages
Practicals 1-2. Caesar Cipher

Shin-Cheng Mu

July 2018

1. Go to our course homepage and download `CaesarCipher.zip`.
2. The main task is to define the functions below:

$$\begin{aligned} &\mathit{encode} :: \mathit{Int} \to \mathit{String} \to \mathit{String} \quad , \\ &\mathit{crack} \quad :: \mathit{String} \to \mathit{Int} \quad , \\ &\mathit{decode} :: \mathit{String} \to \mathit{String} \quad , \end{aligned}$$

   such that *encode k xs* enciphers *xs* using the key k, *crack ys* takes a ciphered string and tries to recover the key, and *decode xs* deciphers the input string (using *crack*).
3. Many auxiliary functions are currently given as "undefined". You may need to define your own auxiliary functions too.
4. This practical is adapted from a chapter in Hutton [Hut07]. For many fascinating stories about cryptography, see Singh [Sin00].

References

[Hut07] Graham Hutton. *Programming in Haskell*. Cambridge University Press, 2007.

[Sin00] Simon Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor, 2000.