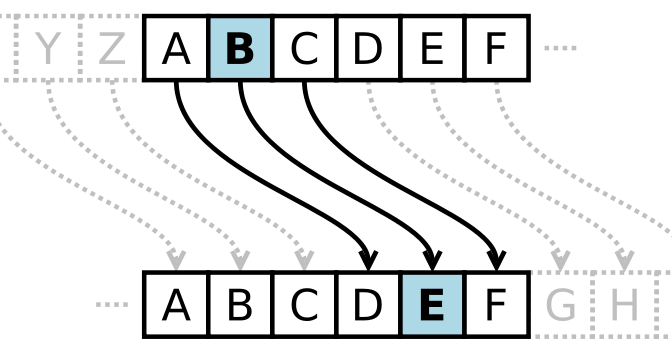
# The Caesar Cipher – Description and Requirements

The Caesar Cipher is an ancient method of encrypting text, i.e. to transform text into a format that is unreadable for anyone without a secret key. It is believed that Julius Caesar actually used such a cipher for his correspondence. Unfortunately for him this type of cipher is easily broken using a frequency analysis method which is outlined below. Your assignment is to implement this in Java.

### **Part 1: Encrypting and Decrypting**

The Caesar cipher is a “rotation cipher” and operates by translating each letter into the one that is shifted along the alphabet by a fixed distance. This distance is called the shift. It is the same for all letters in the alphabet and therefore can be seen as the secret key to encrypt and decrypt: To encrypt your text using a given shift, you translate letters by that many places later in the alphabet. A Caesar cipher with shift 3 can be illustrated as follows.



For example, if your text to encrypt is “Meet me at midnight under the bridge” and your shift is 3, the encrypted text is “Phhw ph dw plgqljkw xqghu wkh eulgjh” because the letter ‘M’ gets translated into a ‘P’ and ‘e’ gets translated into ‘h’ and so on. We “wrap around” at the end of the alphabet so that ‘z’ gets changed to a ‘c’ given a shift of 3. We can interpret a negative value for the shift as translating letters backwards (e.g. an ‘f’ gets encrypted as the letter ‘b’ if the shift is -4).

#### **Requirements**

In a file called Caesar.java, implement the following (public static) methods.

* a method called rotate that rotates a single character. It should have two arguments: the first is an integer (int) shift and the second a char to rotate, and return the character rotated by the given shift, as a char.

Lower-case characters should be translated into lower-case characters, capitalised ones into capitalised ones, and all other characters should remain untouched.

* another method called rotate that rotates a whole string. It should again take two arguments: the first is an integer (int) shift and the second a String to rotate. It should return the string rotated by the given shift, as a String. Of course you may use your character rotation method here.
* a main method, that allows to encode/decode text.

This should read both the shift as well as the plain text from command line arguments. The first argument is interpreted as the shift and the second one as a (string) message to be rotated. Your method should check if it was called with exactly two arguments and complain otherwise. The only output printed by the program should be the rotated string output. See below for example outputs. Your program needs to print the exact same output to be considered correct.

$> java Caesar 3 "The ships hung in the sky in much the same way that bricks don't."

Wkh vklsv kxqj lq wkh vnb lq pxfk wkh vdph zdb wkdw eulfnv grq'w.

$> java Caesar -13 "The ships hung in the sky in much the same way that bricks don't."

Gur fuvcf uhat va gur fxl va zhpu gur fnzr jnl gung oevpxf qba'g.

$> java Caesar 13 The ships hung in the sky in much the same way that bricks don't.

Too many parameters!

Usage: java Caesar n "cipher text"

$> java Caesar 13

Too few parameters!

Usage: java Caesar n "cipher text"