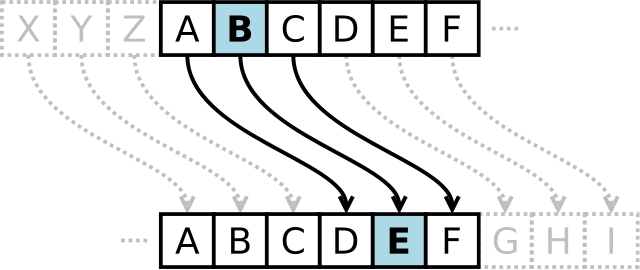
Activity 1: Caesar Cipher

This activity requires an algorithmic solution and corresponding script to implement a Caesar cipher. This is a simple encryption cipher that uses the 26 uppercase letters of the English alphabet. Given a specific encryption key, the Caesar cipher shifts the letters in the plaintext (the unencrypted message) so that they are mapped to new letters in the ciphertext (the encrypted message).

As an example, imagine that the encryption key is the number +3. This means that each letter in the plaintext is shifted three places to the right:



Wikipedia, <https://en.wikipedia.org/wiki/Caesar_cipher>

The letter ‘B’ becomes the letter ‘E’, the letter ‘D’ becomes the letter ‘G’ etc. For the letters near the end of the English alphabet, we ‘wrap around’ as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plaintext letter | U | V | W | X | Y | Z |
| Ciphertext letter | X | Y | Z | A | B | C |

In this way, all 26 letters in the plaintext are mapped to 26 different letters in the ciphertext.

Here is an example of a plaintext message encrypted by the Caesar cipher with a shift of +3:

|  |  |
| --- | --- |
| Plaintext (spaces removed): | MEETMEINTHEHOTELLOBBYATTHREEPM |
| Ciphertext: | PHHWPHLQWKHKRWHOOREEBDWWKUHHSP |

The spaces in the plaintext are removed because they cannot be ciphered.

To decrypt this message, we apply a negative shift of the same distance, i.e. -3, to shift each letter 3 places to the left, revealing the original plaintext:

**HINT:** Convert all letters, in the original message, to upper case. Convert full stop ‘.’ to the letter ‘X’ (as ‘X’ is rare, we can use it to separate sentences).

**Challenges:** Dealing with input such as the following string:

4000 ‘Words... $#^!&\*%! words, we hate them!!!’?

The solution must:

Accept the plaintext string

Prepare/convert plaintext as required

Accept cipher key

Generate cipher

Capture and report potential errors

Generate ciphertext