Exercise 4.1.1 Affine Cipher

Task

Crack the Affine Cipher

```
KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI
```

Solution

Using the background knowledge that the text is in French but we can still use the English letter frequencies we start off by counting the occurrences of all letters:

```
C: 32
B: 21
K: 20
P: 20
I: 16
...
```

The letter `C` is the most common so we can assume it's the letter `E`. The next most common letter in English is `T` so let's check if `T` gets encoded to `B`.

We can try to break the Affine cipher with the above two assumptions:

$$\begin{cases} \mathcal{E}(E) = C \\ \mathcal{E}(T) = B \end{cases}$$

or by using the mapping letter $\rightarrow$ number:

Letter	**A**	**B**	**C**	**D**	**E**	**...**	**T**	**...**	**Z**
Number	0	1	2	3	4	...	19	...	25

$$\begin{cases} \mathcal{E}(4) = 2 \\ \mathcal{E}(19) = 1 \end{cases}$$

where $y = \mathcal{E}(x) = ax + b \pmod{26}$ is the Affine cipher encryption function.

Substituting in the above function we get

$$\begin{cases} 2 = 4a + b \pmod{26} \\ 1 = 19a + b \pmod{26} \end{cases}$$

Subtracting the first equation from the 2nd we get

$$15a = -1 = 25 \pmod{26}$$

Using the Extended Euclidean algorithm (or Wolfram Alpha) we can find that the inverse of a is 7. By multiplying both sides with 7 we find that

$$a = 19 \pmod{26}$$

Substituting a in $2 = 4a + b \pmod{26}$ gives us the equation

$$4 \times 19 + b = 2 \pmod{26}$$

Using the same method as before we get the result

$$a \equiv 19, b \equiv 4$$

Hence the encryption function is $\mathcal{E}(x) = 19x + b \pmod{26}$.

To find the decryption function $\mathcal{D}(x) = a^{-1}(x - b) \pmod{26}$ we need to find a^{-1}.

We can find it by solving $19 \times a^{-1} = 1 \pmod{26}$ which gives us $a^{-1} \equiv 11$.

Hence the decryption function is $\mathcal{D}(x) = 11(x - 7) \pmod{26}$.

Using the decryption function on the encrypted text we get:

```
OCANADATERREDENOSAIEUXTONFRONTESTCEINTDEFLEURONSGLORIEUXCARTONBRASSAITPORT
ERLEPEEILSAITPORTERLACROIXTONHISTOIREESTUNEEPOPEEDESPLUSBRILLANTSEXPLOITSE
TTAVALEURDEFOITREMPEEPROTEGERANOSFOYERSETNOSDROITS
```

Which seems to be the French lyrics of the Canadian national anthem.

Code used for this exercise

To find the number of occurrences of each letter

```
from collections import Counter

input = """KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI"""

print(Counter(input))
```

To decrypt the message

```
import string

input = """KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI"""

# Clean up new lines
input = [x for x in input if x != '\n']

result = ''
a_inverse = 11
b = 4
for x in input:
    # Map [A-Z] -> [0-26]
    x = ord(x) - ord('A')

    y = (a_inverse * (x - b)) % 26

    # Map [0-26] -> [A-Z]
    result += chr(y + ord('A'))

print(result)
```