# FKLIIUH#GH#FHVDU



### Objectif

L'objectif de ce projet est de décrypter la phrase suivante:

PRZRFFNTRARPBAGVRAGEVRAQVAGRERFFNAGZNVFVYRFGFHSSVFNZZRAGYBATCBHEARCNFYRQRPELCGRENYNZNVA

Le principe est simple: chaque lettre de la phrase d'origine (le message clair) a été décalé d'un certain nombre de rangs dans l'alphabet, toujours le même. Ce nombre est la *clé* de chiffrement.

Comme il n'y a que 25 décalages possibles, il vous faudra tous les tester un par un (méthode par bruteforce) pour réussir le déchiffrement.

#### Pré-requis

En cryptographie, les méthodes sont très souvent numériques. Plutôt que de manipuler les caractères eux-mêmes, on préfère manipuler leurs représentations dans un encodage (on étudiera cette question un peu plus tard dans l'année).

Avec Python, on peut récupérer le code Unicode d'un caractère avec la fonction ord :

## **%** Script Python

```
>>> ord('A')
65
```

et le caractère correspondant à un code Unicode avec la fonction chr :

#### **%** Script Python

```
>>> chr(65)
'A'
```

# E Pour vous guider

1

Définissez une fonction decale(lettre) qui décale de 3 rangs dans l'alphabet la lettre majuscule lettre passée en argument (après Z, on recommencera à A...)

2

Rajoutez un paramètre n à la fonction précédente pour pouvoir décaler la lettre de n rangs.

3

Utilisez la fonction précédente pour créer la fonction decale\_phrase(p, n) qui décale toutes les lettres d'une phrase p de n rangs.

### Ouverture

Écrire une fonction qui crypte une chaine de caractères avec au choix:

- le chiffre de Vigenère;
- la méthode du masque jetable;

#### Proposition de correction

1.

```
def decale_lettre(lettre: str) -> str:
1
2
3
        Renvoie la lettre située 3 places plus loin dans l'alphabet,
4
        en revenant éventuellement au début.
5
6
        rang = ord(lettre) + 3
7
        if rang > ord('A') + 25:
8
            rang = rang - 26
9
        return chr(rang)
```

2.

```
1
    def decale_lettre(lettre: str, n: int) -> str:
2
3
        Renvoie la lettre située n places plus loin dans l'alphabet,
4
        en revenant éventuellement au début.
        1.1.1
5
6
        rang = ord(lettre) + n
7
        if rang > ord('A') + 25:
8
            rang = rang - 26
9
        return chr(rang)
```

3.