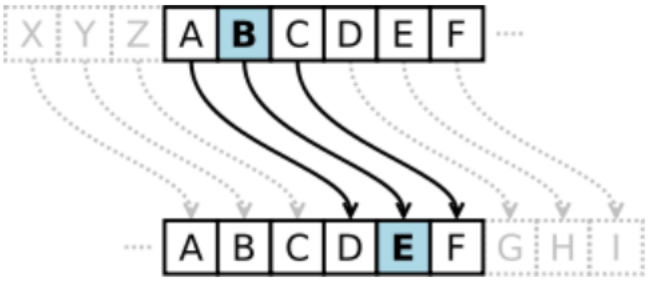
TP cryptographie avec Python

Exercice 1 : Le code de César

C'est une méthode ancienne de cryptographie qui consiste à réaliser un décalage constant dans l'ordre alphabétique. Ce mode de cryptographie a été rapidement abandonné car une fois qu'on connaît la méthode, le décryptage est très simple.



1) Créer un script nommé `cesar.py` et définir tout d'abord une variable globale `alphabet` ainsi :

```
alphabet = ['a', 'b', 'c', 'd', 'e', 'f',
            'g', 'h', 'i', 'j', 'k', 'l',
            'm', 'n', 'o', 'p', 'q', 'r',
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```

2) Créer une fonction `decaler_alphabet` qui prendra en paramètre un entier `n` qui représente le décalage à effectuer. La fonction renverra une nouvelle liste dont les lettres auront subi un décalage de n positions par rapport à la liste globale `alphabet`

```
>>> liste_decalee = decaler_alphabet(3)
>>> liste_decalee
['d', 'e', 'f', ...]
```

- On créera tout d'abord dans la fonction une copie indépendante de `alphabet`
- On pourra utiliser ensuite les méthodes `pop` et `append` pour compléter cette nouvelle liste à partir de `alphabet`

syntaxe	description
`liste.pop(0)`	renvoie le premier élément de liste et le supprime de liste

Le principe est donc d'enlever le premier élément de la liste avec `pop` puis de l'insérer à la fin avec `append` et ceci n fois (n étant le décalage).

3) Créer une fonction `chiffrer_lettre` qui prendra deux paramètres :

- `lettre` : la lettre à chiffrer
- `alphabet_decale` : la liste décalée à partir de alphabet

La valeur de retour sera la nouvelle lettre chiffrée correspondant au caractère passé en paramètre.
Pour y arriver il faut trouver l'indice de `lettre` dans la variable globale `alphabet` renvoyer la lettre contenue au même indice dans `alphabet_decale`

AIDE : On pourra utiliser la méthode `index`

syntaxe	description
`liste.index(x)`	renvoie l'indice de la première occurrence de x dans liste

4) Créer la fonction `chiffrer_mot` qui chiffre les lettres d'un `mot` passé en paramètre et renvoie le mot chiffré. Il sera possible d'utiliser la fonction `chiffrer_mot` dans le programme principal :

```
Entrée : aucune
Sortie : aucune

1:  cle ← Demander le décalage souhaité
2:  alphabet_decale ← Décaler l'alphabet du décalage souhaité
3:  clair ← Demander le mot à chiffrer
4:  chiffre ← Chiffrer le clair, i.e chiffrer chaque lettre du mot
5:  Afficher chiffre
```

Remarque : L'alphabet utilisé nous empêche d'utiliser des lettres capitales ou accentuées dans le mot à chiffrer !

5) Créer la fonction `dechiffrer_mot`, basé sur le même principe que la fonction `chiffrer_mot`, à la différence qu'elle déchiffre le `mot` passé en paramètre.

6) Combien existe-t-il de clés permettant d'obtenir des alphabets décalés différents ? En déduire une implémentation d'une fonction `decrypter_mot` prenant en entrée un `mot` chiffré et renvoyant l'ensemble des mots clairs possibles.
*Remarque : Pour décrypter, nous ne connaissons pas à priori la clé utilisée. D'où le fait de tester par la méthode **brute force**, i.e tester toutes les possibilités.*

Exercice 2 : Utilisation d'un masque jetable

Pour rappel, la fonction booléenne OU EXCLUSIF(XOR) se définit par la table de vérité ci-dessous.

a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

1) Réaliser l'opération XOR, en complétant la ligne `chiffré` du tableau suivant.

`clair`		1	0	0	1	1	0	1	0
`masque`	⊕	1	0	1	0	1	0	0	1
`chiffré`	=								

2) Que vaut l'expression $(\text{clair} \oplus \text{masque}) \oplus \text{masque}$?

3) Créer une fonction `chaine_to_utf(chaine)` qui renvoie un tableau des codes ASCII des caractères de la `chaine`.

AIDE : La fonction `ord` renvoie le code ASCII d'un caractère. La fonction `chr` fait l'opération inverse.

4) On considère la variable suivante :

```
masque = "CETTEPHRASEESTVRAIMENTTRESTRESLONGUEMAISCESTFAITEXPRES"
```

Créer une fonction `chiffre_dechiffre(message, masque)` qui chiffre `message` en le XORant avec `masque`.

AIDE : L'opérateur XOR s'écrit « ^ » et permet de réaliser l'opération directement sur des **entiers**.

*Remarque : Cette fonction doit pouvoir **aussi** servir à déchiffrer le message chiffré.*