TD 2 et TP 2 et 3

Chiffrement Symétrique

Consigne importante valable pour toutes vos productions ;

Pour pouvoir faire fonctionner les tests des docString de chaque fonction, ajouter au début « du code main » :

```
if  __name__== "__main__":
  import doctest
  doctest.testmod()
```

Toute les fonctions produites en Info0603 devront comporter au moins un test rédigé **avant** l'écriture du code. Cette pratique permet ainsi d'avancer sa réflexion tout en documentant le code.

C'est aussi un bon canal de communication avec son binôme ou le professeur.

Note : pour gagner du temps, reprenez les fichiers du moodle et remplacez par du code les `raise NotImplementedError`.

Déja fait : Premiers chiffreurs

Ecrire la classe ChiffreurParDecalage durant le TD, compléter son code durant le TP.

Ecrire la classe ChiffreurVigenere durant le TD, compléter son code durant le TP.

Proposer une méthode pour déchiffrer les documents DocChiffre1, DocChiffre2, DocChiffre3, DocChiffre4 puis les déchiffrer.

Les attaques par fréquences, demandées au TP1 pourront être faits en fin de TP3 si nécessaire mais tout le monde doit commencer le TP2 par l'exercice suivant.

Exercice 1: Chiffreur Affine :

Ecrire à la main les sorties du programme suivant :

```
monBin=Binaire603([0x00,0x01,0x02,0x010,0x20,0x40,0x80])
for monCodeur in [ChiffreurAffine(3,5), ChiffreurAffine(1,1),
                          ChiffreurAffine(1,0), ChiffreurAffine(2,5)]:
  print(f"Codage avec monCodeur :")
  print(" Bin:",monBin)
  monBinC=monCodeur.binCode(monBin)
  print(" Bin Codé:",monBinC)
  monBinD=monCodeur.binDecode(monBinC)
  print(" Bin Décodé:",monBinD)
  print(" monBinD (décodé) est égal à Monbin ?",monBinD==monBin)
```

Quel est l'espace des clés ? Son cardinal ?

Comparer le chiffrement par décalage avec le chiffrement affine

Ecrire le code de cette classe : on gagnera à utiliser ElmtZnZ.

Tester sur un Binaire603 mais aussi un Texte603 courts.

Ecrire un algorithme d'attaque de ce chiffre.