Le DM est à rendre pour le **28 mai 2017** au plus tard par mail à l'adresse alexandrebenoit@yahoo.fr.
En cas de soucis, on peut me contacter par mail.
L'objectif de ce DM est de manipuler des textes.
On verra deux problèmes différents et indépendants. On écrira chaque exercice dans un fichier différent (Il y aura donc 2 fichiers) sous la forme `NOM_dm_ex1.py` et `NOM_dm_ex2.py`.
L'objectif de l'exercice 1 est de rechercher un mot dans un texte et de le modifier. On verra comment modifier tous les mots du texte.
L'objectif de l'exercice 2 est de coder et décoder un texte, on verra comment attaquer un code, c'est à dire trouver la clé qui a codé le message.
Pour tester les fonctions, on pourra manipuler des fichiers texte avec l'aide des fonctions disponible dans **dm.py**.

Exercice 1 : Recherche d'un mot

1. Écrire la fonction **recherche_mot** qui prend en arguments deux listes de caractères (**texte** et **mot**) et renvoie la position du mot dans le texte ou -1 si le mot n'y est pas.
   Exemple :
   ```
   >>>texte = list('Je suis en BCPST en 1ère année')
   >>>mot = list('en')
   >>>position = recherche_mot(texte, mot)
   >>>print(position)
   8
   ```
2. Écrire la fonction **modifier_mot** qui prend en arguments deux listes de caractères (**texte** et **mot**) et un entier (**indice**) et renvoie le texte où l'on a changé le mot à la position **indice** du texte par **mot** (on considère que les mots sont de même taille)
   Exemple :
   ```
   >>>texte = list('Je suis en BCPST en 1ère année')
   >>>mot = list('ne')
   >>>texte = modifier_mot(texte, mot, 8)
   >>>print(texte)
   ['J','e',' ','s','u','i','s',' ','n','e',' ','B','C','P','S','T',' ','e','n',
   ' ','1','è','r','e',' ','a','n','n','é','e']
   ```
3. Écrire la fonction **chercher_et_remplacer** qui prend en arguments trois listes de caractères (**texte**, **mot1** et **mot2** avec **len(mot1)=len(mot2)**) et qui renvoie le texte où l'on a modifié la première occurrence de **mot1** par **mot2**, cette fonction renverra le texte inchangé si l'élément n'apparait pas.
   Exemple :
   ```
   >>>texte = list('Je suis en BCPST en 1ère année')
   >>>mot1 = list('en')
   >>>mot2 = list('ne')
   >>>texte = chercher_et_remplacer(texte, mot1, mot2)
   >>>print(texte)
   ['J','e',' ','s','u','i','s',' ','n','e',' ','B','C','P','S','T', ','e','n',','
   ' ','1','è','r','e',' ','a','n','n','é','e']
   ```

4. Écrire la fonction **chercher_et_remplacer_tout** qui prend en arguments trois listes de caractères (**texte**, **mot1** et **mot2** avec **len(mot1)=len(mot2)**) et qui renvoie le texte où l'on a modifié toutes les occurrences de **mot1** par **mot2**.
   Exemple :

```
>>>texte = list('Je suis en BCPST en 1ère année')
>>>mot1 = list('en')
>>>mot2 = list('ne')
>>>texte = chercher_et_remplacer_tout(texte, mot1, mot2)
>>>print(texte)
['J','e',' ','s','u','i','s',' ','n','e',' ','B','C','P','S','T', ','n','e',','
' ','1','è','r','e',' ','a','n','n','é','e']
```

5. Écrire la fonction **modifier_mot2** qui prend en arguments deux listes de caractères (**texte** et **mot**) et deux entiers (**indice** et **taille**) et renvoie le texte où l'on a changé le mot à la position **indice** et de longueur **taille** du texte par **mot** (on ne considère plus que les mots sont de même taille)
   Exemple :

```
>>>texte = list('Je suis en BCPST en 1ère année')
>>>mot = list('encore en')
>>>texte = modifier_mot(texte, mot, 8, 2)
>>>print(texte)
['J','e',' ','s','u','i','s',' ','e','n','c','o','r','e',' ','e','n',' ',
'B','C','P','S','T',' ','e','n',' ','1','è','r','e',' ','a','n','n','é','e']
```

6. Écrire la fonction **chercher_et_remplacer_tout2** qui prend en arguments trois listes de caractères (**texte**, **mot1** et **mot2**) avec des longueurs quelconques) et qui renvoie le texte où l'on a modifié toutes les occurrences de **mot1** par **mot2**.
   Exemple :

```
>>>texte = list('Je suis en BCPST en 1ère année')
>>>mot1 = list('en')
>>>mot2 = list('none')
>>>texte = chercher_et_remplacer_tout(texte, mot1, mot2)
>>>print(texte)
['J','e',' ','s','u','i','s',' ','n','o','n','e',' ','B','C','P','S',
'T',' ','n','o','n','e',' ','1','è','r','e',' ','a','n','n','é','e']
```

   On pourra faire aussi en sorte d'avoir le résultat suivant :

```
>>>texte = list('Je suis en BCPST en 1ère année')
>>>mot1 = list('en')
>>>mot2 = list('encore en')
>>>texte = chercher_et_remplacer_tout(texte, mot1, mot2)
>>>print(texte)
['J','e',' ','s','u','i','s',' ','e','n','c','o','r','e',' ','e','n',' ','B','C','P','S',
'T',' ','e','n','c','o','r','e',' ','e','n',' ','1','è','r','e',' ','a','n','n','é','e']
```

Exercice 2 : Codage et décodage d'un texte

Le but de cet exercice est de crypter et décrypter un texte par le chiffrement de César. Le texte chiffré s'obtient en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet (le principe est clairement expliqué sur Wikipedia)

En informatique, chaque caractère est codé par un entier (voir par exemple Wikipedia).

En Python, la fonction **chr** permet de convertir un entier en un caractère et la fonction **ord** permet de convertir un caractère en un entier.

Par exemple :

```
>>>chr(80)
'P'
>>>chr('P')
80
```

1. Écrire la fonction **cesar** qui prend en argument une liste de texte (**texte**) et une clé (**cle**) et décale chaque caractère du texte de **cle**

   Exemple :

   ```
   >>>texte = list('BCPST')
   >>>cle = 5
   >>>texte = cesar(texte, cle)
   >>>print(texte)
   ['G', 'H', 'U', 'X', 'Y']
   ```

2. Écrire la fonction inverse **dechiffrer** qui prend en argument une liste de texte (**texte**) et une clé (**cle**) et décode le texte chiffré par **cesar** et la clé (**cle**)

On souhaite maintenant attaquer ce code.

On sait que la lettre la plus fréquente en Français est le **e**. L'idée est de déterminer le caractère le plus fréquent dans le texte et de le comparer avec **e**.

On verra qu'en pratique le caractère le plus fréquent est l'espace. On adaptera donc notre algorithme.

3. Caractère le plus fréquent.
   (a) Écrire la fonction **chercher** qui prend en argument une liste de caractères (**texte**) et un caractère (**element**) et qui renvoie la position de la première occurence d'**element** s'il est dans le texte et -1 sinon.
   (b) Écrire la fonction **nombre_apparitions** qui prend en argument une liste de caractères (**texte**) et renvoie deux listes **L1** qui contient tous les caractères de **texte** et **L2** qui contient le nombre d'occurrences de chaque caractère dans le texte. Exemple :

   ```
   >>>texte = list('Je suis en BCPST en 1ère année')
   >>>L1, L2 = nombre_apparitions(texte)
   >>>print(L1)
   ['J', 'e', ' ', 's', 'u', 'i', 'n', 'B', 'C', 'P', 'S', 'T', '1', 'A', 'è',
   'r', 'a', 'é']
   >>>print(L2)
   [1, 5, 6, 2, 1, 1, 4, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1]
   ```

   (c) Écrire une fonction **tri** (on pourra adapter une des fonctions vu en cours) qui prend en argument une liste de caractères (**L1**) et une liste de nombres (**L2**) et tri par ordre décroissant **L1** et **L2** en fonction de **L2**. Exemple avec **L1** et **L2** des exercices précédents :

   ```
   >>>tri(L1,L2)
   >>>print(L1)
   [' ', 'e', 'n', 's', '1', 'J', 'u', 'i', 'B', 'C', 'P', 'S', 'T', 'A', 'è', 'r',
    'a', 'é']
   >>>print(L2)
   [6, 5, 4, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
   ```

4. Déduire de la question précédente une fonction qui permet d'attaquer le codage de César.