

Pour faire ce TD, vous aurez besoin de recherche personnelle en ligne.

## 1 Encodage

1. **Combien de caractères peut-on encoder en :**
  - a) ascii
  - b) latin-1 (iso 8859-1)
  - c) cp1252 (windows)
2. **Peut-on encoder le caractère € en :**
  - a) ascii
  - b) latin-1
  - c) cp1252
  - d) utf-8
3. **En utf-8 les points de code Unicode sont encodés sur une taille de 1 à 4 octets. Et en utf-32, tous les points de code sont encodés sur 4 octets.**
  - (a) Trouver un avantage de utf-8 par rapport à utf-32.
  - (b) Trouver un avantage de utf-32 par rapport à utf-8.
4. **Un ordinateur "reçoit" le message suivant : 010011100101001101001001**  
Décoder le message en latin-1.
5. **En utf-8, le symbole € est encodé sur 3 octets : E282AC.**  
Sur un ordinateur avec le système d'exploitation Linux, on enregistre un fichier texte avec ce seul caractère (encodé par défaut en utf-8).  
Puis on ouvre ce fichier sur un vieil ordinateur avec un système d'exploitation Windows 98 pour lequel l'encodage par défaut est cp1252.
  - (a) Le fichier va-t-il pouvoir s'ouvrir dans un éditeur de texte ?
  - (b) Si oui, que va-t-il s'afficher ?

## 2 Texte et Python

Les exercices suivants peuvent se traiter à l'aide d'une console Python.

1. **Points de code et encodage utf8**
  - (a) Donner le point de code des caractères suivants : "à", "â", et "€".  
Sur combien d'octets chacun de ces caractères est-il représenté en UTF-8 ?
  - (b) Quel est le caractère de point de code 128013 ?  
Sur combien d'octets ce caractère est-il représenté en UTF-8 ?
2. **Alphabet**
  - (a) Écrire une succession d'instructions utilisant une boucle, permettant d'affecter à la variable `alphabet` la chaîne "abcdefghijklmnopqrstuvwxy".
  - (b) Un pangramme est une phrase comportant toutes les lettres de l'alphabet.  
*Exemple* : "Portez ce vieux whisky au juge blond qui fume."  
Écrire une fonction nommée `est_pangramme` prenant en paramètre une chaîne de caractères et renvoyant `True` si cette chaîne est un pangramme et `False` sinon.

```
>>> est_pangramme("pangramme")
False
>>> est_pangramme("Portez ce vieux whisky au juge blond qui fume")
True
```

```
>>> est_pangramme("the quick brown fox jumps over the lazy dog")
True
>>> est_pangramme("Dans un wagon bleu, tout en mangeant cinq kiwis frais,
vous jouez du xylophone")
True
```

### 3. Chiffrement

Le Code de César, ou chiffrement par décalage, est une méthode de chiffrement de message.

Écrire une fonction `cesar(texte, n)` qui renvoie le message `texte` chiffré avec la méthode de chiffrement par décalage de clé `n`.

On supposera que `texte` ne contient pas de caractère accentué. Le message chiffré ne contiendra plus d'espace ou ponctuation, et tous ses caractères seront en lettres capitales.

```
>>> cesar("Quand tout est permis, rien n'est possible", 3)
'TXDQGWXRXWHVWSHUPLVULHQHVWSRVVLEOH'
>>> cesar("Quand tout est permis, rien n'est possible", 7)
'XBHUKAVBALZAWLYTPZYPLUJLZAWVZZPISL'
>>> cesar("Quand tout est permis, rien n'est possible", 26)
'QUANDTOUTESTPERMISRIENNESTPOSSIBLE'
```

### 4. Palindrome

Un palindrome est un texte qui peut se lire indifféremment de gauche à droite, ou de droite à gauche. On ne tient compte ni des accents, ni des espaces.

Écrire une fonction `est_palindrome` qui renvoie `True` si le texte passé en paramètre est un palindrome et `False` sinon.

```
>>> est_palindrome("palindrome")
False
>>> est_palindrome("radar")
True
>>> est_palindrome("Engage le jeu que je le gagne")
True
```