Remarques

— Les fonctions ne devront pas utiliser les méthodes natives fournies par le langage Python, telles que *max(), min()*.

— On attend une *docstring* pour chaque fonction.

— Pour les exercices 1 à 5 on utilisera, pour tester les fonctions, un tableau de 30 notes comprises entre 0 et 20 (que l'on construira par compréhension).

Exercice 1 : Écrire la fonction `extrema(tab: list) → tuple` qui renvoie les notes minimale et maximale du tableau *tab*, sous forme de tuple.

Exercice 2 : Écrire la fonction `maxi_position(tab: list) → tuple` qui renvoie la note maximale du tableau *tab* et l'indice de la première apparition de cette note, sous forme de tuple.

Exercice 3 : Écrire la fonction `maxi_position_dernier(tab: list) → tuple` qui renvoie la note maximale du tableau *tab* et l'indice de la dernière apparition de cette note, sous forme de tuple.

Exercice 4 : Écrire la fonction `maxi_nb(tab: list) → int` qui renvoie le nombre d'apparitions de la note maximale du tableau *tab*.

Exercice 5 :

1. Écrire la fonction `est_present(tab: list, note: int) → bool` qui renvoie `True` si `note` est présent dans `tab`.
2. De quel paramètre dépend la durée d'exécution de la fonction ? Peut-on faire mieux ?

Exercice 6 : On peut assimiler une chaîne de caractère à un tuple, c'est à dire une séquence ordonnée et non modifiable. Ainsi on peut repérer un caractère par son indice.

```
1 mot = "bonjour"
2 print(mot[3]) # affiche 'j'
```

1. Écrire la fonction `est_voyelle(lettre: str) → bool` qui renvoie `True` si `lettre` est une voyelle, `False` sinon.
2. Écrire la fonction `compter_voyelles(mot: str) → dict` qui renvoie un dictionnaire du décompte des voyelles. On utilisera un dictionnaire `voyelles` qui associe chaque voyelle à l'entier 0.

```
1 voyelles = {"a": 0, "e": 0, "i": 0, "o": 0, "u": 0, "y": 0}
```

3. Dans le programme principal, écrire la boucle qui affiche dans la console chaque voyelle suivie de son nombre d'occurrences. Par exemple, on pourra afficher :
   — A : 3
   — E : 5
   — …
4. Écrire la fonction **max_voyelles(voyelles : dict) → list** qui renvoie les voyelles qui comptent le plus d'occurrences dans le dictionnaire renvoyé par la fonction précédente. Par exemple, pour le mot "orangeade" la fonction `compter_voyelles` renvoie le dictionnaire suivant :

```
1 {'a': 2, 'e': 2, 'i': 0, 'o': 1, 'u': 0, 'y': 0}
```

   Ainsi le tableau renvoyé par la fonction `max_voyelles` sera ['a', 'e'].

Christophe Viroulaud