Semaine 4 : mercredi 25/09 au mercredi 2/10

Exercice 1 :
Écrire cinq fonctions `extremum_v.`, prenant en paramètre un tableau d'entiers `tab`, et qui renvoie :

- `version 1` : le plus grand entier présent dans `tab`.
- `version 2` : l'indice de la première occurrence du plus grand entier présent dans `tab`.
- `version 3` : le couple correspondant à l'indice de la dernière occurrence du plus grand entier présent dans `tab` ainsi que cette valeur maximale.
- `version 4` : le couple correspondant à la valeur minimale de `tab` et la liste de tous les indices de son apparition dans `tab`.
- `version 5` : un dictionnaire dont les clés sont `'minimum'` et `'maximum'`. La valeur associée à chaque clé correspond à l'entier correspondant à l'information de la clé au sein du tableau `tab`. Lorsque le tableau est vide, renvoie le dictionnaire `{'minimum': None, 'maximum': None}`

Exemples :

```
>>> extremum_v1([-4, 8, 7, 8, -1, 2, 8, -1, 5])
8
>>> extremum_v1([5, -1, 0, 12, 9, 2, 2, 6])
12
>>> extremum_v2([-4, 8, 7, 8, -1, 2, 8, -1, 5])
1
>>> extremum_v2([5, -1, 0, 12, 9, 2, 2, 6])
3
>>> extremum_v3([-4, 8, 7, 8, -1, 2, 8, -1, 5])
(6, 8)
>>> extremum_v3([5, -1, 0, 12, 9, 2, 2, 6])
(3, 12)
>>> extremum_v4([4, -8, 7, -8, 1, -2, -8, 1, -5])
(-8, [1, 3, 6])
>>> extremum_v5([-4, 8, 7, 8, -1, 2, 8, -1, 5])
{'minimum': -4, 'maximum': 8}
>>> extremum_v5([-2, -5, -4, -1, -4, -7, -2])
{'minimum': -7, 'maximum': -1}
>>> extremum_v5([])
{'minimum': None, 'maximum': None}
```

Exercice 2 :
Les résultats d'un vote ayant plusieurs candidats sont stockés dans un tableau. On ne connaît pas à l'avance le nombre de candidats, ni leurs noms.
Par exemple, avec les candidats `'Romane'`, `'Aziz'` et `'Léa'` :

```
urne = ['Romane', 'Aziz', 'Romane', 'Léa', 'Léa', 'Aziz', 'Léa', 'Aziz', 'Romane', 'Aziz',
 ↪  'Aziz', 'Léa', 'Léa']
```

Écrire une fonction `depouiller`, prenant en paramètre un tableau `urne` contenant les résultats d'un vote, et qui renvoie un dictionnaire dont les clés sont les noms des candidats et les valeurs le nombre de votes en leur faveur.

Par exemple :

```
>>> voix = depouiller(urne)
>>> voix
{'Romane': 3, 'Aziz': 5, 'Léa': 5}
```

Écrire une fonction `maximum_voix`, prenant en paramètre un dictionnaire `voix` correspondant aux comptages des voix, et qui renvoie le couple correspondant au tableau des candidats ayant eu le maximum de voix et ce maximum de voix. Le tableau renvoyé sera construit par compréhension.

Par exemple :

```
>>> maximum_voix(voix)
(['Aziz', 'Léa'], 5)
```

Exercice 3 :
Une phrase est constituée de mots, constitués de lettres, séparés par un espace, ou une apostrophe, ou une virgule suivie d'un espace. Une phrase se termine par un point accolé au dernier mot, ou par un point d'exclamation ou d'interrogation possédant un espace avec le dernier mot.
Remarque : il existe d'autres symboles pour constituer une phrase, que ceux cités. Dans cet exercice, seulement ces cas particuliers seront considérés.

Exemples de phrases acceptées :
- `Voici un premier exemple.`
- `Un espace est présent à la fin avec le point d'exclamation !`
- `Lorem ipsum dolor sit amet, consectetur adipiscing elit.`

Écrire une fonction `nb_mots`, prenant en paramètre une chaîne de caractères `phrase`, et qui renvoie le nombre de mots dans la `phrase`.
Remarque : on pourra faire le lien entre le nombre d'espaces ou d'apostrophes et le nombre de mots.

Exemples :

```
>>> nb_mots('Voici un premier exemple.')
4
>>> nb_mots("Un espace est présent à la fin avec le point d'exclamation !")
12
>>> nb_mots('Lorem ipsum dolor sit amet, consectetur adipiscing elit.')
8
```

Exercice 4 :
L'ordre des gènes sur un chromosome est représenté par un tableau `ordre` de `n` cases d'entiers distincts deux à deux et compris entre `1` et `n`.

Par exemple, `ordre = [5, 4, 3, 6, 7, 2, 1, 8, 9]` dans le cas `n = 9`.

On dit qu'il y a un point de rupture dans `ordre` dans chacune des situations suivantes :
- la première valeur de `ordre` n'est pas `1` ;
- l'écart entre deux gènes consécutifs n'est pas égal à `1` ;
- la dernière valeur de `ordre` n'est pas `n`.

Par exemple, si `ordre = [5, 4, 3, 6, 7, 2, 1, 8, 9]` avec `n = 9`, on a
- un point de rupture au début car 5 est différent de 1
- un point de rupture entre 3 et 6 (l'écart est de 3)
- un point de rupture entre 7 et 2 (l'écart est de 5)
- un point de rupture entre 1 et 8 (l'écart est de 7)

Il y a donc 4 points de rupture.

Écrire une fonction `est_un_ordre`, prenant en paramètre un tableau d'entier `tab`, et qui renvoie `True` si `tab` représente bien un ordre de gènes de chromosome et `False` sinon.

Exemples :

```
>>> est_un_ordre([1, 6, 2, 8, 3, 7])
False
>>> est_un_ordre([5, 4, 3, 6, 7, 2, 1, 8, 9])
True
```

Écrire une fonction `nombre_points_rupture`, prenant en paramètre un tableau `ordre` représentant un ordre des gènes, et qui renvoie le nombre de points de rupture d'un tableau passé en paramètre représentant l'ordre de gènes d'un chromosome.
On écrira une assertion au début du corps de la fonction vérifiant que le tableau passé en paramètre est bien un ordre.

Exemples :

```
>>> nombre_points_rupture([5, 4, 3, 6, 7, 2, 1, 8, 9])
4
>>> nombre_points_rupture([1, 2, 3, 4, 5])
0
>>> nombre_points_rupture([1, 6, 2, 8, 3, 7, 4, 5])
7
>>> nombre_points_rupture([2, 1, 3, 4])
2
```