Exercices : Structure de données linéaires

Exercice 1

Écrire une fonction recherche_min qui prend en paramètre un tableau de nombres non trié t, et qui renvoie l'indice de la première occurrence du minimum de ce tableau. Les tableaux seront représentés sous forme de liste Python.

Quelques exemples :

recherche_min([5])
0
recherche_min([2,4,1])
2
recherche_min([5,3,2,2,4])
2

Exercice 2

On considère la fonction separe ci-dessous qui prend en argument un tableau tab dont les éléments sont des 0 et des 1 et qui sépare les 0 des 1 en plaçant les 0 en début de tableau et les 1 à la suite.

```
def separe(tab):
    i = 0
    j = ...
    while i < j:
        if tab[i] == 0:
            i = ...
        else:
            tab[i], tab[j] = ...
            j = ...
    return tab
```

Compléter la fonction separe ci-dessus.

Quelques exemples :

separe([1,0,1,0,1,0,1,0])
[0,0,0,0,1,1,1,1]
separe([1,0,0,0,1,1,0,1,1,0,1,0,1,1,1,0])
[0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1]

Exercice 3

Sur le réseau social **Tip Top**, on s'intéresse au nombre de "like" des abonnés. Les données sont stockées dans des dictionnaires où les clefs sont les pseudos et les valeurs correspondantes sont les nombres de "like" comme ci-dessous :

```
{'Bob': 102, 'Ada': 201, 'Alice': 103, 'Tim': 50}
```

Écrire une fonction max_dico qui :

- Prend en paramètre un dictionnaire dico non vide dont les clés sont des chaines de caractères et les valeurs associées sont des entiers ;
- Renvoie un tuple dont :
  - La première valeur est la clé du dictionnaire associée à la valeur maximale ;
  - La seconde valeur est la première valeur maximale présente dans le dictionnaire.

Quelques exemples :
max_dico({'Bob' :102,'Ada' :201,'Alice' :103,'Tim' :50})
('Ada',201)
max_dico({'Alan' :222,'Ada' :201,'Eve' :220,'Tim' :50})
('Alan',222)

Exercice 4

On souhaite manipuler des dates en Python. Il existe un module Date mais on ne va l'utiliser.

On considère qu'une date contient le jour, le mois et l'année. Par exemple, aujourd'hui nous sommes le 6 septembre 2022. On s'intéressera aux dates comprises entre les années 2000 et 2100.

L'objectif est de :

— créer des variables qui ont pour valeur une date.

— créer des opérations sur les dates comme l'égalité ou la comparaison ;

— déterminer si une date est vide ou non ;

— déterminer si une date est valide ;

— afficher une date en utilisant une chaine de caractères

1) Donner différents types permettant de créer une date. On fera plusieurs propositions en précisant le type utilisé.

2) La fonction est_vide(d) prend en paramètre une date d et renvoie un booléen affirmant si la d est vide ou non. Écrire le code de cette fonction.

3) La fonction est_valide(d) prend une date d en paramètre et renvoie un booléen qui affirme si la date d est correctement écrite, dans le bon type et le bon format et avec des valeurs cohérentes. Écrire le code de cette fonction.

4) La fonction egales prend en paramètres deux dates et renvoie un booléen affirmant que les dates saisies sont les mêmes. Écrire le code de cette fonction.

5) La fonction est_avant(d1,d2) prend en paramètres deux dates d1 et d2 et renvoie un booléen affirmant que la date d1 est plus ancienne que la date d2. Écrire le code de cette fonction.

6) Créer une fonction d'affichage de date.