

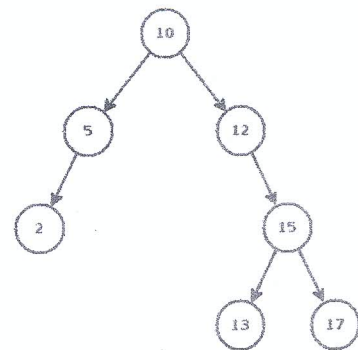
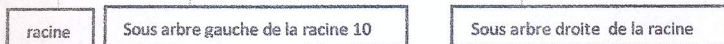
Devoir de contrôle n°2**Matière : INFORMATIQUE****Classe : 1^{ère} Année MP – PT – PC Durée : 1h***Toutes les réponses doivent être rédigées en Python.**L'indentation dans les instructions Python sera prise en considération dans le barème***Exercice 1**

Soit la structure de donnée suivante nommée Arbre binaire.
 Cette structure est représentée en Python avec une liste de la forme :

[racine , sousArbreGauche, sousArbreDroite]

Exemple pour la figure ci-dessus :

A1 = [10, [5, [[2,[],[]] , []], [12, [], [15, [[13,[],[]], [17,[],[]]]]]]]



- A1 est une liste dont :

- le premier élément 10 de A1 représente le nœud racine de l'arbre, de type entier.
- le deuxième élément de A1 représente le sous arbre gauche de A1, de type liste.
- le troisième élément de A1 représente le sous arbre droite de A1, de type liste.

- Le sous arbre gauche de A1 est une liste dont le premier élément est sa racine, le deuxième élément est son sous arbre gauche, le troisième est son sous arbre droite. Pareil pour le sous arbre droit de A1.

- Les nœuds 2, 13 et 17 sont des feuilles. Une feuille est un arbre ayant une racine, n'ayant ni de sous arbre gauche, ni sous arbre droite.

- les valeurs des nœuds du sous Arbre gauche sont inférieurs à la racine

- les valeurs des nœuds du sous arbre droite sont supérieurs à la racine

Travail demandé :

1. Ecrire une fonction `est_vide` prenant en paramètre un arbre A et qui vérifie si A est vide.
Exemple: `est_vide ( [] )` retourne True.
2. Ecrire une fonction `val_racine(A)` qui retourne la valeur de la racine de l'arbre A.
Cette fonction traite l'exception `IndexError` dans le cas où A est vide et retourne None.
Exemple: `val_racine ([ 5, [ [2,[],[]] , [] ] )` retourne 5.
3. Ecrire une fonction `est_feuille(A)` qui vérifie si A est une feuille.
Exemple : `est_feuille([2,[],[]])` retourne True.
4. Ecrire une fonction **réursive** `hauteur(A)` qui retourne la hauteur de l'arbre A en tant que le maximum entre la hauteur du sous arbre gauche et la hauteur du sous arbre droite, sachant que la hauteur d'une feuille égale à 1.
Exemple: `hauteur(A1)` retourne 4
5. Ecrire une fonction **réursive** `taille(A)` qui calcule et retourne la taille de l'arbre A en tant que le nombre total de nœuds.
Exemple: `taille(A1)` retourne 7
6. Ecrire une fonction **réursive** `setsortedList(A)` qui retourne une liste triée par des valeurs des nœuds de l'arbre A en triant récursivement le sous arbre gauche puis la racine ensuite le sous arbre droit.
Exemple: `setSortedList(A1)` retourne [2, 5, 10, 12, 13, 15, 17]

Exercice 2 :

Un gouvernement souhaite suivre la propagation d'une épidémie dans différentes villes du pays pour une année donnée. On vous demande de développer un programme en Python permettant de gérer ces données avec un dictionnaire dont :

- les clés sont les noms des villes
- les valeurs sont des dictionnaires ayant deux couples de la forme :

{ 'population' : nombre de population , 'infectes' : une liste de 12 éléments ayant le nombre des infectés pour chaque mois }

Exemple :

```
dict_ville = {
    "Sfax": {
        "population": 1000000,
        "infectes": [120, 150, 180, 210, 250, 300, 350, 400, 380, 320, 280, 240]
    },
    "Djerba": {
        "population": 400000,
        "infectes": [80, 100, 120, 140, 160, 180, 200, 220, 210, 190, 170, 150]
    },
    "Mednine": {
        "population": 600000,
        "infectes": [90, 110, 130, 150, 170, 200, 230, 250, 240, 220, 200, 180]
    }
}
```

- Par exemple :
 - o il y a 80 cas infectés en mois de Janvier (01) dans la ville Djerba
 - o Il y a 240 cas infectés en mois de décembre (12) dans la ville Sfax

Travail demandé :

1. Ecrire une fonction **taux_infection(dict_ville, ville, mois)** qui renvoie le pourcentage des infectés en ce mois par rapport à la population de la ville. Cette fonction traite l'exception ZeroDivisionError dans le cas où on divise par une population égale à 0. Dans ce cas la fonction retourne le message 'La population est 0, impossible de calculer le taux d'infection.'

Exemple : `taux = taux_infection(dict_ville, 'mednine', 1)` # `taux = 100 * ( 90 / 600000)`

2. Ecrire une fonction **mois_critique(dict_ville, ville)** qui renvoie le numéro du mois où le nombre des infectés est maximum pour la ville **ville**. Cette fonction gère l'exception KeyError dans le cas où la ville passée en paramètre n'existe pas dans le dictionnaire **dict_ville** en affichant le message d'erreur noté dans l'exemple.

Exemple : `mois_critique(dict_ville, 'Sfax')` retourne 8

`mois_critique(dict_ville, 'Alger')` retourne "Alger n'existe pas dans les villes"

NB : `lst.index(val)` retourne l'indice de la première occurrence val de la liste lst

3. Ecrire une fonction **ville_plus_touchee(dict_ville, mois)** qui renvoie le nom de la ville ayant un nombre d'infecté maximum pour le mois donné **mois**.

Exemple : `ville_plus_touchee(dict_ville, 3)` retourne 'Sfax'

4. Ecrire une fonction **récursive total_infections_toutes_villes(liste_infectes)** qui renvoie le total des infectés de toutes les villes durant les 12 mois.

Exemple : `total_infections_recuratif(dict_ville['Sfax']['infectes'])` retourne 3180