Ex1

def somme\_feuilles(a):

# Ici le cas de base, ca veut dire l'enfant d'une feuille, puisque une feuille

# n'a pas d'enfant retourner 0

if len(a) == 0:

return 0

# si il s'agit d'une feuille [val,[],[]]

# le noeud n'a pas d'enfant gauche [] et d'enfant droit []

elif len(a[1]) + len(a[2]) == 0:

return a[0]

else:

# Sinon retourner le nombre d'enfants de sous arbre gauche a[1] et sous arbre droit a[2]

return somme\_feuilles(a[1]) + somme\_feuilles(a[2])

a2 = [19, [42, [], [17, [10, [], []], []]], [15, [35, [], []], [12, [], []]]]

print(somme\_feuilles(a2))

ex2

niveau = -1

def memeniveau(a, l):

    global niveau

    # arbre est vide

    if len(a) == 0:

        return True

    # si il s'agit d'une feuille [val,[],[]]

    # le noeud n'a pas d'enfant gauche [] et d'enfant droit []

    elif len(a[1]) + len(a[2]) == 0:

        if niveau < 0:

            niveau = l

            return True

        return l == niveau

    else:

        return memeniveau(a[1], l+1) and memeniveau(a[2], l+1)

a2 = [19, [42, [], [17, [], []]], [15, [35, [], []], [12, [], []]]]

print(memeniveau(a2, 0))

ex3

# Une fonction utilitaire pour obtenir la somme des valeurs

# dans un arbre avec la racine "a"

def somme(a):

    # Enfant d'une feuille !

    if a == []:

        return 0

    else:

        # somme des valeurs à gauche+racine+valeurs à droite

        return somme(a[1])+a[0]+somme(a[2])

# renvoie True si la propriété SommeEnfant est

# vraie pour le nœud donné et ses deux enfants

def SommeEnfant(a):

    # Si le nœud est vide [], retournez true

    if a == []:

        return True

    # S'il s'agit d'un nœud feuille, retournez true

    if a[1] == [] and a[2] == []:

        return True

    else:

        # somme des éléments de sous arbre gauche

        gauche = somme(a[1])

        # somme des éléments de sous arbre droit

        droit = somme(a[2])

        # si le nœud et ses deux enfants satisfont la propriété,

        # retourne True sinon False

        if a[0] == (gauche+droit) and

            (SommeEnfant(a[1]) == True and SommeEnfant(a[2]) == True):

            return True

        return False

a0 = [17, [6, [3, [3, [], []], []], []], [5, [], []]]

print(SommeEnfant(a0))