

## TP 06

## Fonctions récursives

*Savoirs et compétences :*

- ☐ AA.C9 : Choisir un type de données en fonction d'un problème à résoudre
- ☐ AA.S12 : Fichiers

## Proposition de corrigé

## Activité 1 – Introduction et définition

Activité 2 – Programmation de la factorielle  $n!$ 

**Question 1** Proposer une fonction `fact_it(n:int)->int` renvoyant la factorielle de  $n$  par une méthode itérative

```
def fact_it(n : int) -> int :
    res = 1
    for i in range (1,n+1): # ou range(n)
        res = res*i
    return res
```

**Question 2** Proposer une fonction `fact_rec(n:int)->int` renvoyant la factorielle de  $n$  par une méthode récursive

```
def fact_rec(n : int) -> int :
    if n == 0 :
        return 1
    else :
        return n * fact_rec(n-1) # Récursion
```

## Activité 3 – Recherche dichotomique

**Question 3** Écrire une fonction `dichotomie_rec(x0,L)` qui renvoie True ou False selon que  $x_0$  figure ou non dans  $L$  par cette méthode. On utilisera une boucle `while` que l'on interrompra soit lorsque l'on a trouvé  $x_0$ , soit lorsque l'on a fini de parcourir la liste.

```
def appartient_dicho_rec(e : int , t : list) -> bool:
    """Renvoie un booléen indiquant si e est dans t
    Préconditions : t est un tableau de nombres trié par ordre croissant e est un nombre"""
    g = 0 # Limite gauche de la tranche où l'on recherche e
    d = len(t)-1 # Limite droite de la tranche où l'on recherche e
    while g <= d: # La tranche où l'on cherche e n'est pas vide
        m = (g+d)//2 # Milieu de la tranche où l'on recherche e
        pivot = t[m]
        if e == pivot: # On a trouvé e
            return True
        elif e < pivot:
            d = m-1 # On recherche e dans la partie gauche de la tranche
```

```

    appartient_dicho_rec(e,t[g:d])
else:
    g = m+1 # On recherche e dans la partie droite de la tranche
    appartient_dicho_rec(e,t[g:d])
return False

```

## Activité 4 – Recherche de palindrome

### Question 4

```

def palindrome(x):
    if len(x)<=1:
        return True
    elif x[0]==x[-1]:
        return palindrome(x[1:-1])
    else:
        return False

```

## Activité 5 – Bubble bobble

**Question 5** Définir la fonction `bubble1(n:int, x:float, y:float, r:float) -> None` permettant de tracer la figure suivante.

```

def bubble1(n, x=0, y=0, r=8):
    circle([x, y], r)
    if n > 1:
        bubble1(n-1, x+3*r/2, y, r/2)
        bubble1(n-1, x, y-3*r/2, r/2)

```

**Question 6** Définir la fonction `bubble2(n:int, x:float, y:float, r:float, d:str) -> None` permettant de tracer la figure suivante.

```

def bubble2(n, x=0, y=0, r=8, d=''):
    circle([x, y], r)
    if n > 1:
        if d != 's':
            bubble2(n-1, x, y+3*r/2, r/2, 'n')
        if d != 'w':
            bubble2(n-1, x+3*r/2, y, r/2, 'e')
        if d != 'n':
            bubble2(n-1, x, y-3*r/2, r/2, 's')
        if d != 'e':
            bubble2(n-1, x-3*r/2, y, r/2, 'w')

```