

Exercice 1 3 points

On a essayé d'écrire une fonction récursive `pair(n)`, dont le but est de déterminer si un nombre n est pair ou non.

```
1 def pair(n):
2     if n == 0 :
3         return True
4     else :
5         return pair(n-2)
```

1. Quelle critique peut-on émettre à propos de ce code ?
2. Proposer une amélioration de ce code.

Exercice 2 3 points

On considère la fonction `myst(lst,k)` ci-dessous :

```
1 def myst(lst, k):
2     if k == -1 :
3         return 0
4     else :
5         return lst[k] + myst(lst, k-1)
```

1. Calculer `myst([1,5,2,10,3],3)`.
2. Quelle condition sur le paramètre `k` faudrait-il rajouter pour éviter que cette fonction ne provoque des erreurs ?

Exercice 3 3 points

On rappelle que le PGCD de deux nombres entiers a et b vérifie les propriétés suivantes :

- $\text{pgcd}(a,b) = \text{pgcd}(b,r)$, où r est le reste de la division euclidienne de a par b .
- $\text{pgcd}(b,0) = b$

Écrire le code Python d'une fonction récursive `pgcd(a,b)` qui renvoie le PGCD de deux nombres entiers a et b .

Exercice 4 5 points

On rappelle que la suite de Fibonacci est définie de la manière suivante :

- $F_0 = 0$ et $F_1 = 1$
 - pour tout $n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$
1. Écrire une fonction récursive `fibonacci(n)` qui renvoie la valeur de F_n pour tout entier naturel n passé en paramètre.
 2. Dessiner les différents états de la pile d'appel lors de l'exécution de l'instruction `fibonacci(3)`.

Exercice 5 6 points

La fonction récursive d'Ackermann (1896-1962, Allemagne), notée Ack dans la suite de l'exercice, est définie par :

$$\text{pour } m \in \mathbb{N}, n \in \mathbb{N}, \quad Ack(m,n) = \begin{cases} n+1 & \text{si } m=0 \\ Ack(m-1,1) & \text{si } m>0 \text{ et } n=0 \\ Ack(m-1, Ack(m,n-1)) & \text{si } m>0 \text{ et } n>0 \end{cases}$$

1. Calculer les valeurs suivantes.
a. $Ack(0,1)$ b. $Ack(0,2)$ c. $Ack(1,0)$ d. $Ack(2,0)$ e. $Ack(1,2)$
2. Écrire le code Python de la fonction `ack(m,n)` qui renvoie le nombre $Ack(m,n)$.