

Sujet n°19

Exercice1 :

```
def multiplication(n1,n2):
    m=0
    if n1<0:
        n1=-n1
        n2=-n2
    for i in range(n1):
        m=m+n2
    return m
```

if n1<0: n1=-n1 n2=-n2	Si n1 est un nombre négatif Alors on transforme n1 et n2 en nombre négatif et vu que n1 est déjà un nombre négatif il devient alors un nombre positif la même pour n2
for i in range(n1): m=m+n2	Permet de faire l'opération n1*n2 mais que avec des addition

Exercice 2 :

```
def chercher(T,n,i,j):
    if i < 0 or j > len(T) :
        print("Erreur")
        return None
    if i > j :
        return None
    m = (i+j) // 2
    if T[m] < n :
        return chercher(T, n, i+1 , j)
    elif T[m] > n :
        return chercher(T, n, i , j-1)
    else :
        return m
```

if i < 0 or j > len(T)	i est le premier indice de recherche dans le tableau, le premier indice d'un tableau étant 0, i ne peut pas être inférieur. j est le dernier indice de recherche dans le tableau, il ne peut pas être supérieur à la taille du tableau, on utilise donc len().
if i > j	Le premier indice ne peut pas être supérieur au dernier.

$m = (i+j) // 2$	On crée une valeur m milieu des indices de recherche dans le tableau, 'coupant' le tableau en deux parties. On utilise deux « / » pour que le résultat soit un entier.
<pre> if T[m] < n : return chercher(T, n, i+1, j) elif T[m] > n : return chercher(T, n, i, j-1) </pre>	On compare l'indice m du tableau (on n'oublie pas les []) avec l'entier recherché n, étant donné que le tableau est rangé par ordre croissant : -s'il est inférieur, alors n se trouve dans la seconde partie. On rappelle donc la fonction en réduisant la taille de la première (on utilise i+1 pour commencer la recherche à un indice plus élevé). -s'il est supérieur, alors n se trouve dans la première partie. On rappelle donc la fonction en réduisant la taille de la seconde (on utilise j-1 pour terminer la recherche à un indice moins élevé). A force de réduire le tableau de recherche, on retrouvera n s'il existe dans les indices de recherche.
<pre> else : return m </pre>	Si l'indice m n'est ni inférieur ni supérieur : -m est égal à n donc on return m. -n n'existe pas dans le tableau donc on return une valeur vide.