# Corrigé sujet 33 - Année : 2022

**Sujet 33 - 20222**

## 1. Exercice 1

```python
def convertir(T):
    poids =  len(T)-1
    valeur = 0
    for elt in T:
        valeur += 2**poids * elt
        poids -=1
    return valeur
```

**Commentaires**

On peut aussi effectuer un parcours par indice (cela invite d'utiliser la variable `poids`)

## 2. Exercice 2

```
1   def tri_insertion(L):
2       n = len(L)
3
4       # cas du tableau vide
5       if L==[]:
6           return L
7
8       for j in range(1,n):
9           e = L[j]
10          i = j
11
12      # A l'etape j, le sous-tableau L[0,j-1] est trie
13      # et on insere L[j] dans ce sous-tableau en determinant
14      # le plus petit i tel que 0 <= i <= j et L[i-1] > L[j].
15          while  i > 0 and L[i-1] > L[j]: #(1)
16              i = i - 1
17
18          # si i != j, on decale le sous tableau L[i,j-1] d'un cran
19          # vers la droite et on place L[j] en position i
20          if i != j:
21              for k in range(j,i,-1): #(2)
22                  L[k] = L[k-1]
23              L[i] = e #(3)
24      return L
```

1. On se contente de suivre les indications données en commentaire.
2. On décale les éléments de façon à laisser libre l'emplacement d'indice `i`.
3. On a sauvegardé dans `e` la valeur à insérer.

**Attention**

1. On peut regretter les noms de variables courts et donc fort peu explicites.
2. L'insertion dans le début de liste se fait souvent en échangeant le nombre avec son voisin de droite tant qu'il lui est inférieur (ou que le début de liste n'est pas atteint)
3. Le commentaire ligne 18 parle du "*sous tableau L[i,j-1]*", il faut comprendre les éléments du tableau dont les indices sont entre `i` et `j-1`. De même ligne 12 pour le *sous tableau L[0,j-1]*.
4. Cette fonction fait un tri en place et modifie donc la liste `L`, le `return` final laisse cependant croire qu'on a construit un nouveau table qu'on souhaite renvoyer.