

Sujet 4

❑ Exercice 1 : Conversion du binaire vers le décimal

Écrire une fonction `binaire_vers_decimal` qui prend en paramètre une chaîne de caractères composée uniquement de "0" et de "1" et renvoie l'entier ayant cette chaîne de caractères comme écriture binaire. Par exemple, `binaire_vers_decimal("1001011")` renvoie 75. En effet : $2^6 + 2^3 + 2^1 + 2^0 = 75$.

Exemples :

```
1 >>> binaire_vers_decimal("1000000")
2 64
3 >>> binaire_vers_decimal("1111010")
4 122
5 >>> binaire_vers_decimal("11001100")
6 204
```

❑ Exercice 2 : Séparation des 0 et des 1 dans une liste

On souhaite écrire une fonction `separe` qui prend en argument une liste composée uniquement de 0 et de 1 et modifie cette liste de façon à ranger les 0 en début de liste et les 1 à la fin. Par exemple, `separe([1,0,0,1,0,1,0])` doit renvoyer `[0,0,0,0,1,1,1]`. Pour cela on propose l'algorithme suivant :

1. on crée deux indices `i` et `j`, tous les nombres situés *avant* `i` sont des 0 et tous ceux situés *après* `j` sont des 1.
2. Tant que `i < j` certaines valeurs ne sont pas classées, si la valeur est un 0, on décale l'indice `i` à droite sinon c'est un 1 et donc on l'échange avec l'indice `j` et on diminue `j`

```
1 def separe(liste):
2     i = ....
3     j = ....
4     while .... :
5         if liste[i] == ....:
6             i = ....
7         else:
8             liste[i], liste[j] =
9                 j = .....
```

Exemple :

```
1 >>> liste = [0,1,1,1,0,0,1,0,1]
2 >>> separe(liste)
3 >>> liste
4 [0,0,0,0,1,1,1,1,1]
```