

Exercice 1 : Combien y a-t-il de façons d'échanger une somme s en utilisant des billets ou des pièces ayant les valeurs suivantes : 100, 50, 10, 5 et 1 euro ?

Par exemple, 90 euros peuvent être échangés de plusieurs manières :

- $50 + 4 \times 10$
- 9×10
- $8 \times 10 + 10 \times 1$, etc.

On suppose que les types de billets ou pièces possibles sont stockés dans une liste `[1, 5, 10, 50, 100]` et que :

- si $s = 0$ on compte 1.
- si $s < 0$ on compte 0.

1. Écrire une fonction récursive qui calcul le nombre de façons d'échanger une somme s .

Exercice 2 : On peut déterminer combien de chiffres possède un entier positif en effectuant des divisions entières par 10 jusqu'à ce que le quotient devienne nul. Le nombre de chiffres correspond alors au nombre de divisions effectuées.

1. Écrire une fonction récursive qui calcul le nombre de chiffres dans un entier positif.

Exercice 3 : Lorsque vous copiez une liste `l` en utilisant `l.copy()`, seule une copie de la liste est créée. La nouvelle liste pointera vers les mêmes éléments que `l`, ce qui peut entraîner un comportement inattendu lorsque ces éléments sont des structures mutables comme une liste de listes.

La méthode `copy.deepcopy` effectue également une copie récursive de telles listes récursives. La différence est illustrée par le code suivant :

```
import copy
l = [[1], [2]]
lc = l.copy()
ldc = copy.deepcopy(l)
l[0][0] = 3
```

1. Copier le code précédent et afficher chacune des listes `lc`, `l` et `ldc`.
2. Écrire une fonction récursive qui crée une copie récursive d'une liste imbriquée.

Remarque : Une liste peut être référencée plusieurs fois et il peut exister des références cycliques parmi les listes comme dans l'exemple suivant :

```
l = [[1], [2], [3]]
l[2] = l
```

Pour gérer la copie de telles structures récursives, il est important de noter que chaque liste `l` possède un identifiant unique `id(l)`.