

Correction du TP 1

Partie I

question 1

```
In [1]: #définition de la caisse
caisse=[200,100,50,20,10,5,2,1]
caisse
```

```
Out[1]: [200, 100, 50, 20, 10, 5, 2, 1]
```

question 2

```
In [2]: def rendmonnaie(s,c):
        n=len(c)
        rendu=len(c)*[0]
        somme_a_rendre=s*100
        i=0
        while somme_a_rendre>0 and i<=n-1:
            while somme_a_rendre>=c[i] :
                somme_a_rendre=somme_a_rendre-c[i]
                rendu[i]=rendu[i]+1
            i=i+1
        return rendu
```

Tests

```
In [3]: rendmonnaie(2.34,caisse)
```

```
Out[3]: [1, 0, 0, 1, 1, 0, 2, 0]
```

```
In [4]: rendmonnaie(5.71,caisse)
```

```
Out[4]: [2, 1, 1, 1, 0, 0, 0, 1]
```

```
In [5]: #autre version
def rendmonnaiebis(s,c):
    n=len(c)
    rendu=len(c)*[0]
    somme_a_rendre=s*100
    i=0
    while somme_a_rendre>0 and i<=n-1:
        if somme_a_rendre>=c[i] :
            rendu[i]=int(somme_a_rendre//c[i])
            somme_a_rendre=somme_a_rendre%c[i]
        i=i+1
    return rendu
```

Tests

```
In [6]: rendmonnaiebis(2.34,caisse)
```

```
Out[6]: [1, 0, 0, 1, 1, 0, 2, 0]
```

```
In [7]: rendmonnaiebis(5.71,caisse)
```

```
Out[7]: [2, 1, 1, 1, 0, 0, 0, 1]
```

Partie II

```
In [8]: def rendmonnaie2(s,c):  
        n=len(c)  
        rendu=[[c[i],0] for i in range(len(c))]  
        somme_a_rendre=s*100  
        i=0  
        while somme_a_rendre>0 and i<=n-1:  
            while somme_a_rendre>=c[i] :  
                somme_a_rendre=somme_a_rendre-c[i]  
                rendu[i][1]=rendu[i][1]+1  
            i=i+1  
        return rendu
```

Tests

```
In [9]: rendmonnaie2(2.34,caisse)
```

```
Out[9]: [[200, 1], [100, 0], [50, 0], [20, 1], [10, 1], [5, 0], [2, 2], [1, 0]]
```

```
In [10]: rendmonnaie2(5.71,caisse)
```

```
Out[10]: [[200, 2], [100, 1], [50, 1], [20, 1], [10, 0], [5, 0], [2, 0], [1, 1]]
```

```
In [11]: #version bis  
def rendmonnaie2bis(s,c):  
    n=len(c)  
    rendu=[[c[i],0] for i in range(len(c))]  
    somme_a_rendre=s*100  
    i=0  
    while somme_a_rendre>0 and i<=n-1:  
        if somme_a_rendre>=c[i] :  
            rendu[i][1]=int(somme_a_rendre//c[i])  
            somme_a_rendre=somme_a_rendre%c[i]  
        i=i+1  
    return rendu
```

```
In [12]: rendmonnaie2bis(2.34,caisse)
```

```
Out[12]: [[200, 1], [100, 0], [50, 0], [20, 1], [10, 1], [5, 0], [2, 2], [1, 0]]
```

```
In [13]: rendmonnaie2bis(5.71,caisse)
```

```
Out[13]: [[200, 2], [100, 1], [50, 1], [20, 1], [10, 0], [5, 0], [2, 0], [1, 1]]
```

Partie III

question 1

```
In [14]: import random
#Création d'un dictionnaire contenant une quantité aléatoire de pièce
caisse2={caisse[i] : random.randint(0,5) for i in range(len(caisse))}
caisse2
```

```
Out[14]: {200: 0, 100: 2, 50: 4, 20: 1, 10: 4, 5: 0, 2: 4, 1: 0}
```

Question 2

```
In [15]: #Version avec dictionnaire et contrôle de la quantité de pièces disponible
def rendmonnaie3(s,d):
    somme_a_rendre=s*100
    rendu={i:0 for i in d}
    for i in caisse:
        while somme_a_rendre>=i and caisse2[i]>0:
            somme_a_rendre=somme_a_rendre-i
            rendu[i]=rendu[i]+1
            caisse2[i]=caisse2[i]-1
    if somme_a_rendre>0:
        return print("il n'y a plus assez de monnaie")
    else :
        return rendu
```

Remarque : pour le moment, cette partie est codée avec une boucle pour car il y a de très grandes chances que ce soit la solution choisie par les élèves. En effet le recours au dictionnaire rend l'utilisation de la boucle while plus complexe. On pourra pour les élèves plus rapides et donc les plus à l'aise leur demander d'y réfléchir.

Tests

```
In [16]: print("caisse à notre disposition :",caisse2)
print("rendu : ",rendmonnaie3(2.34,caisse2))

caisse à notre disposition : {200: 0, 100: 2, 50: 4, 20: 1, 10: 4, 5: 0, 2: 4, 1: 0}
rendu : {200: 0, 100: 2, 50: 0, 20: 1, 10: 1, 5: 0, 2: 2, 1: 0}
```

```
In [17]: print("caisse à notre disposition :",caisse2)
print("rendu : ",rendmonnaie3(4.54,caisse2))

caisse à notre disposition : {200: 0, 100: 0, 50: 4, 20: 0, 10: 3, 5: 0, 2: 2, 1: 0}
il n'y a plus assez de monnaie
rendu : None
```

```
In [ ]:
```