

Correction exemple 1 Gloutons

```
1 def rendu_monnaie(jeu_pieces , s):
2     """Fonction donnant la liste des pièces et billets à rendre
3     pour obtenir un montant donné et le nombre de pièces ou
4     billets que cela représente.
5     entrées :
6         - jeu_pieces est un tableau (type list) des montants
7         des pièces et billets disponibles pour rendre la monnaie ;
8         - s est un entier positif (type int) représentant la somme
9         à rendre.
10    sorties :
11        - rendu est un tableau (type list) donnant les valeurs des
12        pièces et billets à rendre ;
13        - nb_pieces est un entier positif (type int) indiquant le
14        nombre de pièces ou billets rendus."""
15    rendu = []
16    nb_pieces = 0
17    #indice permettant de trouver le plus grand élément de
18    #jeu_pieces inférieur ou égal à s
19    i = 0
20    while s > 0:
21        while jeu_pieces[i] > s:    #la valeur dépasse s
22            i = i +1                #on regarde la suivante
23        s = s - jeu_pieces[i]
24        rendu.append(jeu_pieces[i])
25        nb_pieces = nb_pieces + 1
26    return rendu, nb_pieces
27
28 #tests
29 jeu = [200, 100, 50, 20, 10, 5, 2, 1]
30
31 assert rendu_monnaie(jeu, 0) == ([], 0)
32 assert rendu_monnaie(jeu, 1) == ([1], 1)
33 assert rendu_monnaie(jeu, 9) == ([5, 2, 2], 3)
34 assert rendu_monnaie(jeu, 27) == ([20, 5, 2], 3)
35 assert rendu_monnaie(jeu, 63) == ([50, 10, 2, 1], 4)
```

```

1 def rendu_monnaie(jeu_pieces, s):
2     """Fonction donnant la liste des pièces/billets à rendre pour
   obtenir
3     un montant donné et le nombre de pièces ou billets que cela
   représente.
4     entrées :
5         - jeu_pieces est un tableau (type list) des montants des pièces
6         et billets disponibles pour rendre la monnaie ;
7         - s est un entier positif (type int) représentant la somme à
   rendre.
8     sorties :
9         - rendu est un tableau (type list) donnant les valeurs des pièces
10        et billets à rendre ;
11        - nb_pieces est un entier positif (type int) indiquant le nombre
12        de pièces ou billets rendus."""
13    rendu = []
14    nb_pieces = 0
15    i = 0
16    while s > 0:
17        while jeu_pieces[i] > s:
18            i = i + 1
19        #on regarde combien de pièces de cette valeur sont à prendre
20        qte = s // jeu_pieces[i]
21        #on obtient le montant restant à rendre
22        s = s % jeu_pieces[i]
23        rendu.append((jeu_pieces[i], qte))
24        nb_pieces = nb_pieces + qte
25    return rendu, nb_pieces
26
27 #tests
28 jeu = [200, 100, 50, 20, 10, 5, 2, 1]
29
30 assert rendu_monnaie(jeu, 0) == ([], 0)
31 assert rendu_monnaie(jeu, 1) == [(1, 1)], 1)
32 assert rendu_monnaie(jeu, 9) == [(5, 1), (2, 2)], 3)
33 assert rendu_monnaie(jeu, 27) == [(20, 1), (5, 1), (2, 1)], 3)
34 assert rendu_monnaie(jeu, 63) == [(50, 1), (10, 1), (2, 1), (1, 1)], 4)

```