

SUJET 13

EXERCICE 2 :

On a ici un algorithme glouton de rendu de monnaie.

```
1 pieces = [1, 2, 5, 10, 20, 50, 100, 200]
2
3 def rendu_monnaie(somme_due, somme_versee):
4
5     rendu = []
6     a_rendre = somme_versee - somme_due
7     i = len(pieces) - 1
8     while a_rendre > 0 :
9         if pieces[i] <= a_rendre :
10            rendu.append(pieces[i])
11            a_rendre = a_rendre - pieces[i]
12        else :
13            i = i - 1
14    return rendu
```

Ligne 5 :

rendu est la variable renvoyée. C'est une liste de nombres.

Je l'initialise ligne 5 avec une liste vide.

Au fur et à mesure de la fonction, on va la remplir avec la valeur des pièces à rendre.

Ligne 6 :

Somme à rendre, de type int (ou float) : c'est un nombre

le calcul à faire est somme_versee – somme_due

Ligne 8 :

Ici on attend une condition d'arrêt.

La fonction doit s'arrêter quand la somme à rendre est atteinte (sous entendu quand il reste 0 dans la variable a_rendre)

Donc la fonction « tourne » Tant que a_rendre > 0

Ligne 10 :

Si la pièce d'indice i peut être rendue :

on ajoute sa valeur à la liste des pièces à rendre

Ligne 11 :

Du coup la somme à rendre diminue de cette valeur

Ligne 13 :

Autre cas : on n'a pas pu rendre la pièce d'indice i, on va regarder si l'on peut rendre la pièces de valeur juste inférieure : l'indice (i) doit diminuer de 1