

Cours NSI	Thème : Algorithmique - Glouton TD – Problème du rendu de monnaie	Date :
-----------	--	--------

1. Contexte

Une caisse automatique d'un supermarché est dotée d'une réserve de pièces et billets.

Lors d'un achat, un client paie en espèces et fournit une certaine somme.

La caisse automatique doit rendre le trop perçu ; pour cela, elle est dotée d'un programme informatique.





Le problème est de déterminer les pièces à rendre de tel sorte que la somme soit rendue avec le **minimum** de pièces possible.


L'objet de ce TD est d'étudier et d'implémenter en python un **algorithme glouton** résolvant le problème du rendu de monnaie.


2. Principe de l'algorithme glouton

L'**algorithme glouton** consiste à répéter le choix de la pièce de plus **grande valeur**, tant qu'elle ne dépasse pas la somme restant à rendre.

 **Question 1** : Soit le système monétaire $S = \{50, 20, 10, 5, 2, 1\}$ et une somme à rendre r de 42 €, Quelles pièces sont à donner selon l'algorithme glouton ci-dessus ?

 **Question 2** : La solution trouvée est-elle optimale ?

 **Question 3** : Soit le système monétaire $S = \{30, 24, 12, 6, 3, 1\}$. Quelles pièces sont à donner pour une somme à rendre r de 49 € selon l'algorithme glouton ci-dessus ?


 **Question 4** : La solution trouvée est-elle optimale ?

Cours NSI	Thème : Algorithmique - Glouton TD – Problème du rendu de monnaie	Date :
-----------	--	--------

3. Implémentation en Python

Soit la spécification et le corps incomplet de la fonction `glouton_rendu`.

```
def glouton_rendu(systeme, a_rendre):
    """
    Détermine les pièces à donner pour rendre une somme selon un système
    monétaire donné
    :param systeme: (list) une collection de n valeurs monétaires, triée par
    ordre décroissant
    :param a_rendre : (int) la somme à rendre
    :return: (list) Un tableau de n valeurs où chaque élément à l'indice i
    correspond au nombres de pièces systeme[i] à rendre.
    """
    n = len(systeme)
    rendu = [0] * n
    for i in range (n):
        if a_rendre .....:
            p = ..... # nombre de pièces de type systeme[i]
            rendu[i] = .....
            a_rendre = .....
    return rendu
```

 Question 5 : Compléter la fonction `glouton_rendu`.

Exemples d'utilisation :

```
>>> glouton_rendu([50, 20, 10, 5, 2, 1], 42)
[0, 2, 0, 0, 1, 0]
>>> glouton_rendu([30, 24, 12, 6, 3, 1], 49)
[1, 0, 1, 1, 0, 1]
```

 Question 6 : Exprimer la complexité de l'algorithme glouton en fonction du nombre de comparaisons.