

```
# -*- coding: utf-8 -*-  
"""
```

```
Created on Wed Dec 7 19:35:29 2022
```

```
@author: dconduche  
"""
```

```
def maximum(L, borne):  
    """Calcule le plus grand terme de L inférieur ou égal à borne  
    maxi = L[0]  
    for x in L[1:]:  
        if x > maxi and x <= borne:  
            maxi = x  
    return maxi
```

```
def renduGlouton(v, S):  
    """ Contenu des variables :  
    v : la somme à rendre  
    S : liste des pièces disponibles, par ordre croissant  
    L : liste des pièces rendues """  
    L = []
```

```
def rendu(v, S):  
    nonlocal L  
    if v >= S[0]: # on peut rendre une pièce (au pire, S[0])  
        piece = maximum(S, v)  
        L.append(piece)  
        rendu(v-piece, S)  
    elif v > 0: # on ne peut pas rendre de pièce, alors c'est une erreur  
        L = "erreur, pas de rendu trouvé"  
    # et sinon, v=0, on sort sans rien faire  
    rendu(v, S) # puis on appelle la fonction locale, comme si on était dans renduGlouton  
    return L
```

```
def renduGloutonIter(v, S):  
    L = []  
    while v >= S[0]:  
        piece = maximum(S, v)  
        L.append(piece)  
        v = v - piece  
    if v == 0:  
        return L  
    else:  
        return "erreur, pas de rendu trouvé"
```

```
S = [1, 2, 5, 10, 20, 50]
#print(renduGlouton(63, S))
L = renduGlouton(63, S)
print(L)
L = renduGloutonIter(63, S)
print(L)
```