CORRECTION
DM 1
INFORMATIQUE

1)
```
def EstSysteme(E):
    if E[-1] != 1:   # le dernier terme doit être 1.
        return False
    for i in range(len(E)-1):
        if E[i] <= E[i+1]:   # ordre non strictement décroissant
            return False
    return True   # Toutes les conditions sont vérifiées.
```

2-a)
```
def PremierIndice(E,s):
    i = 0
    while E[i] > s:   # En sortie de boucle, on a : E[i] <= s
        i += 1
    return i
```

b)
```
def GloutonIt(E,s):
    l = [ ]
    while s > 0:   # Tant qu'il reste une somme à rendre
        i = PremierIndice(E,s)   # on doit rendre E[i].
        l.append(E[i])
        s -= E[i]
    return l
```

c)
```
def GloutonRec(E,s):
    if s == 0:   # cas de base
        return [ ]
    else:
        i = PremierIndice(E,s)
        l = GloutonRec(E, s - E[i])   # Appel récursif
        l.append(E[i])   # on rend E[i]
        return l
```

3 a) Pour rendre s, il faut commencer par choisir une valeur $E[j]$ telle que $E[j] \leq s$ et ensuite rendre la valeur $s - E[j]$. Le nombre minimal de pièces pour rendre s est donc la somme de un et du nombre minimal de pièces permettant de rendre $s - E[j]$, avec la contrainte $E[j] \leq s$.

b)
```
def Min(E,N):
    n = len(E)
    m = [0]   # initialisation avec m[0] = 0.
    for s in range(1, N+1):
        is = PremierIndice(E,s)
        min = m[s - E[is]]   # initialisation du min
        for j in range(is, n):   # recherche du min.
            if m[s - E[j]] < min:
                min = m[s - E[j]]
        m.append(1 + min)
    return m
```