

```
# correction_glouton.py
```

```
001 | ##---- Importation de Modules -----##
002 | from random import randint
003 |
004 |
005 | """
006 | Rendu de monnaie
007 |
008 | """
009 |
010 | # système monétaire
011 | systeme_monetaire =
[1,2,5,10,20,50,100,200]
012 |
013 |
014 | def rendu_glouton(somme_a_rendre,
systeme_monnaie):
015 |     """
016 |     Glouton on rend en premier la pièce de
plus forte valeur
017 |     """
018 |     # liste des pièces à rendre
019 |     lst_pieces = []
020 |     # indice de la première pièce comparer à
la somme à rendre
021 |     i = len(systeme_monnaie) - 1
022 |     while somme_a_rendre > 0:
023 |         valeur = systeme_monnaie[i]
024 |         if somme_a_rendre < valeur:
025 |             i -= 1
026 |         else:
027 |             lst_pieces.append(valeur)
028 |             somme_a_rendre -= valeur
029 |     return lst_pieces
030 |
031 |
032 | #tests de bon fonctionnement
033 | assert rendu_glouton(55,systeme_monetaire) ==
[50,5]
```

```

034| print(rendu_glouton(55,systeme_monetaire))
035| assert rendu_glouton(57,systeme_monetaire) ==
[50,5,2]
036| print(rendu_glouton(57,systeme_monetaire))
037| assert rendu_glouton(58,systeme_monetaire) ==
[50,5,2,1]
038| print(rendu_glouton(58,systeme_monetaire))
039|
040| """
041| Occupation de temps
042|
043| """
044|
045| def generer_intervalles(n):
046|     """
047|     creation d'une liste d'intervalles
048|     """
049|     C = []
050|     for i in range(n):
051|         deb = randint(0,9)
052|         inter = randint(1,10-deb)
053|         fin = deb + inter
054|         C.append([deb,fin])
055|     return C
056|
057| C = generer_intervalles(20)
058|
059| def nommer(C) :
060|     """
061|     Donne des noms aux intervalles
062|     """
063|     for i in range (len(C)):
064|         C[i].append("C_"+str(i))
065|
066| nommer(C)
067|
068|
069| """
070|
071| for i in range(len(C)):

```

```

072 |     print(C [i][1] , end =" ")
073 | print()
074 | """
075 |
076 | #tri des intervalle par valeur croissante de
fin
077 | def echange( t,i,j):
078 |     temp = t[i]
079 |     t[i]= t[j]
080 |     t[j]=temp
081 |
082 | def tri_par_sélection(t):
083 |     """
084 |     Tri la liste en fonction dans l'ordre
croissant des durées de fin
085 |
086 |     """
087 |     for i in range(len(t)-1):
088 |         m = i
089 |         for j in range(i+1,len(t)):
090 |             if t[j][1] <t[m][1] :
091 |                 m = j
092 |             echange(t,i,m)
093 |             # print(t)
094 |
095 | tri_par_sélection(C)
096 |
097 | print(C)
098 |
099 |
100 |
101 | def organiser(liste_intervalle):
102 |     """
103 |
104 |     Etablit une liste des Clients qui peuvent
être successivement servi
105 |     """
106 |     #Tableau du planning initiation
107 |     planning = [liste_intervalle[0]]
108 |     j = 0

```

```

109|     for i in range(1, len(liste_intervalle)):
110|         """
111|         On trouve l'intervalle suivant dont
112|         l'instant de début correspond
113|         au plus près de celui de fin de
114|         l'intervalle précédent.
115|         """
116|         if liste_intervalle[i][0] >=
liste_intervalle[j][1] : # >= pour intervalles
disjoints
117|         planning.append(liste_intervalle[i])
118|         #on passe au suivant
119|         j = i
120|     return planning
121| print(organiser(C))
122|
123|
124|
125| # test avec un liste fixe
126| test = [[0, 2, 'C_14'], [2, 3, 'C_7'], [0, 4,
'C_10'], [0, 4, 'C_12'], [4, 5, 'C_11'], [6, 7,
'C_1'], [2, 7, 'C_3'], [8, 9, 'C_9'], [4, 9,
'C_4'], \
127|         [2, 9, 'C_13'], [1, 9, 'C_15'], [1, 9,
'C_16'], \
128|         [1, 9, 'C_18'], [9, 10, 'C_5'], [9, 10,
'C_0'], [7, 10, 'C_2'], [1, 10, 'C_8'], [9, 10,
'C_17'], [9, 10, 'C_6'], [9, 10, 'C_19']]
129| print(organiser(test))
130| assert organiser(test) == [[0, 2, 'C_14'], [2,
3, 'C_7'], [4, 5, 'C_11'], [6, 7, 'C_1'], [8, 9,
'C_9'], [9, 10, 'C_5']]
131|
132| """
133| Le glouton ne revenant pas en arrière on
remarque un vide de une heure entre C_1 et C_9.
134|

```

135| Le critère du glouton ici étant la fin de
durée des intervalles de temps, les trajets
sélectionnés pour la tournée du camion peuvent
présenté des trous et il ne garanti pas le plus
grands nombre de clients déservis par tournée.

136|

137| ""

138|