

Correction exemple 2 Gloutons

```

1 #Données du problème, sous forme d'une liste de listes, avec le nom
2 #des vidéos (str), leur durée en min (int) et leur taille en Go (float).
3 videos = [{"video1", 114, 4.57}, {"video2", 32, 0.63}, {"video3", 20,
4           1.65},
5           {"video4", 4, 0.085}, {"video5", 18, 2.15}, {"video6", 80,
6           2.71},
7           {"video7", 5, 0.32}]
8
9 def cle_USB(liste_fichiers, taille_max):
10     """Fonction donnant la liste des vidéos à placer sur la clé USB, en
11         utilisant leur durée pour faire les choix.
12         entrées :
13             liste_fichiers est une liste de listes (list), représentant les
14             vidéos disponibles taille_max est un entier positif (int)
15             représentant la place disponible sur le clé en Go
16         sorties :
17             reponse est une liste (list) des vidéos à placer sur la clé USB
18             duree_totale est un entier (int) représentant la durée totale
19             en minutes de ces vidéos"""
20     liste = sorted(liste_fichiers, key = lambda liste_fichiers :
21 liste_fichiers[1], reverse = True)
22     reponse = [] #liste des vidéos à placer sur la clé
23     duree_totale = 0 #durée totale des vidéos placées sur la clé
24     taille_totale = 0 #taille totale des vidéos placées sur la clé
25     i = 0 #indice de parcours de la liste des fichiers
26     disponibles
27     #tant que la taille totale ne dépasse la place disponible sur la clé
28     # et que l'indice ne dépasse pas l'indice maximal de la liste
29     while taille_totale <= taille_max and i < len(liste):
30         nom_video = liste[i][0] #on récupère le nom de la vidéo
31         duree_video = liste[i][1] #on récupère la durée de la vidéo
32         taille_video = liste[i][2] #on récupère la taille de la vidéo
33         #si il reste suffisamment de place pour ajouter cette vidéo
34         if taille_totale + taille_video <= taille_max:
35             reponse.append(nom_video) #on ajoute la vidéo à la liste
36             reponse
37             duree_totale = duree_totale + duree_video #nouvelle durée
38             totale
39             taille_totale = taille_totale + taille_video #nouvelle
40             taille totale
41             i = i + 1 #on regarde l'élément suivant de la liste
42         return reponse, duree_totale
43
44 print(cle_USB(videos, 5))

```

```

1 #Données du problème, sous forme d'une liste de listes, avec le nom des
2 #vidéos (str), #leur durée en minutes (int) et leur taille en Go (float)
3 .
4
5 videos = [ ["video1", 114, 4.57], ["video2", 32, 0.63], ["video3", 20,
6           1.65],
7           ["video4", 4, 0.085], ["video5", 18, 2.15], ["video6", 80,
8           2.71],
9           ["video7", 5, 0.32]]
10
11
12 def cle_USB2(liste_fichiers, taille_max):
13     """Fonction donnant la liste des vidéos à placer sur la clé USB, en
14     utilisant le rapport durée/taille pour faire les choix.
15     entrées :
16         liste_fichiers est une liste de listes (list), représentant les
17         vidéos disponibles
18         taille_max est un entier positif (int) représentant la place
19         disponible sur le clé en Go
20     sorties :
21         reponse est une liste (list) des vidéos à placer sur la clé USB
22         duree_totale est un entier (int) représentant la durée totale en
23         minutes de ces vidéos"""
24     #on classe les fichiers par ordre décroissant des rapports
25     #durée/taille de la vidéo
26     liste = sorted(liste_fichiers, key = lambda liste_fichiers :
27 liste_fichiers[1]/liste_fichiers[2], reverse = True)
28     reponse = []
29     duree_totale = 0
30     taille_totale = 0
31     i = 0
32     while taille_totale <= taille_max and i < len(liste):
33         nom_video = liste[i][0]
34         duree_video = liste[i][1]
35         taille_video = liste[i][2]
36         if taille_totale + taille_video <= taille_max:
37             reponse.append(nom_video)
38             duree_totale = duree_totale + duree_video
39             taille_totale = taille_totale + taille_video
40         i = i + 1
41     return reponse, duree_totale
42
43
44 print(cle_USB2(videos, 5))

```