TPlistechainée_cor.py

```
001|
002| """
003|
004| Liste simplement chainée
005|
006|
007| """
008| import time
009|
010| class Cellule :
011|     def __init__(self,val=None):
012|         self.val = val
013|         self.suivant  = None
014|     def affiche(self):
015|         print(f"Valeur{self.val} position :
{hex(id(self))} suivant : {hex(id(self.suivant))}")
016|
017|
018| class Liste:
019|     #liste simplement chainée
020|
021|     def __init__(self):
022|         self.tete = None
023|
024|     def est_vide(self) :
025|         if self.tete == None :
026|             return True
027|         else:
028|             return False
029|
030|     def taille_liste(self):
031|         # Parcourir toute la liste jusqu'à la
dernière cellule
032|         taille = 0
033|         derniere =  self.tete
034|          # Parcourir toute la liste jusqu'à la
dernière cellule
035|         derniere =  self.tete
```

```
036|         while(derniere):
037|             taille +=1
038|             derniere = derniere.suivant
039|         return(taille)
040|
041|
042|     def inserer_en_tete(self,valeur):
043|         # Créer une nouvelle cellule
044|         # Y inserer la valeur
045|         nouvelle_cellule =Cellule(valeur)
046|         # Mettre la Cellule en tête de la
liste
047|         nouvelle_cellule.suivant =  self.tete
048|         # Faire pointer la liste sur la
cellule
049|         self.tete = nouvelle_cellule
050|
051|     def
inserer_apres(self,cellule_precedente,valeur):
052|         # Vérifier que la cellule exite
053|         if cellule_precedente is None :
054|             print ("La cellule précédente doit
être chainée")
055|             return
056|         # Créer une nouvelle cellule
057|         # Y inserer la valeur
058|         nouvelle_cellule =Cellule(valeur)
059|         #Faire pointer la nouvelle cellule sur
le pointeur de la cellule précédente
060|         nouvelle_cellule.suivant =
cellule_precedente.suivant
061|         #Faire pointer la cellule précedente
sur la nouvelle cellule
062|         cellule_precedente.suivant =
nouvelle_cellule
063|
064|     def
inserer_apres_position(self,position,valeur):
065|         # Vérifier que la position choisie
permette l'insertion
```

```
066|         taille =  self.taille_liste()
067|         if taille < position :
068|             print("Insertion impossible
position supérieure à la taille de la liste ")
069|         elif position  == 0 :
070|             self.inserer_en_tete(valeur)
071|         elif taille == position :
072|             self.inserer_en_queue(valeur)
073|         else :
074|             compteur = 0
075|             # Parcourir toute la liste jusqu'à
la position voulue
076|             derniere =  self.tete
077|             while(compteur < position -1 ):
078|                 compteur +=1
079|                 derniere = derniere.suivant
080|             # Créer une nouvelle cellule
081|             # Y inserer la valeur
082|             nouvelle_cellule =Cellule(valeur)
083|             #Créer une cellule temporaire pour
garder en mémoire la position de la coupure
084|             temporaire = derniere.suivant
085|             #Faire pointer la suite de la
coupure  sur la nouvelle cellule
086|             derniere.suivant =
nouvelle_cellule
087|             #Faire poinyer le suivant de la
nouvelle cellule sur la coupure gardée en mémoire
088|             nouvelle_cellule.suivant =
temporaire
089|
090|
091|     def inserer_en_queue(self,valeur):
092|         # Créer une nouvelle cellule
093|         # Y inserer la valeur
094|         nouvelle_cellule = Cellule(valeur)
095|         # Vérifier si la liste est vide
096|         # Si c'est le cas faire pointer la
tête de liste sur la nouvelle cellule
097|         if self.tete is None :
```

```
098|             self.tete = nouvelle_cellule
099|             return
100|         # Parcourir toute la liste jusqu'à la
dernière cellule
101|         derniere =  self.tete
102|         while(derniere.suivant):
103|             derniere = derniere.suivant
104|         # Ajouter à la dernière cellule de la
liste la cellule créée
105|         derniere.suivant = nouvelle_cellule
106|
107|     def affichage_liste(self):
108|         temporaire = self.tete
109|         print("Valeurs de la liste : ")
110|         while(temporaire):
111|             temporaire.affiche()
112|             temporaire = temporaire.suivant
113|         print("")
114|
115|     def supprimer_une_cellule(self,valeur):
116|         #Stocker la tete de la liste
117|         temporaire = self.tete
118|         #Supprimer la tête dans le cas où la
tête de liste est égale à la valeur
119|         if ( temporaire is not None ):
120|             if(temporaire.val == valeur):
121|                 self.tete = temporaire.suivant
122|                 temporaire = None
123|                 return
124|         #Rechercher la cellule contenant la
valeur
125|         #Stocker la cellule précédente et
sauter la cellule avec la valeur
126|         while(temporaire is not None ):
127|             if temporaire.val == valeur:
128|                     break
129|             precedent = temporaire
130|             temporaire = temporaire.suivant
131|         # La valeur n'est pas dans la liste
132|         if(temporaire == None):
```

```
133|                 return
134|             #Sauter la cellule à supprimer
135|             precedent.suivant = temporaire.suivant
136|             #Supprimer la cellule temporaire
137|             temporaire = None
138|
139|
140| l = Liste()
141|
142| c1 = Cellule(36)
143|
144| l.tete = c1
145|
146|
147|
148| c2 = Cellule(15)
149|
150| c1.suivant = c2
151|
152|
153|
154| c3 = Cellule(0)
155|
156| c2.suivant = c3
157|
158|
159|
160| c4 = Cellule(5)
161|
162| c3.suivant = c4
163|
164|
165|
166| # Ma_Liste = Liste()
167| #
168| # Ma_Liste.inserer_en_tete(1)
169| # Ma_Liste.inserer_en_tete(2)
170| # Ma_Liste.inserer_en_tete(3)
171| # Ma_Liste.inserer_en_tete(4)
172| # Ma_Liste.inserer_en_queue(5)
```

```
# Ma_Liste.inserer_en_queue(6)
# Ma_Liste.affichage_liste()
#
# Ma_Liste.supprimer_une_cellule(3)
#
# Ma_Liste.affichage_liste()
#
# """
# Ma_Liste.inserer_apres_position(0,36)
# Ma_Liste.affichage_liste()
# Ma_Liste.inserer_apres_position(6,36)
# Ma_Liste.affichage_liste()
# Ma_Liste.inserer_apres_position(4,175)
# Ma_Liste.affichage_liste()
# Ma_Liste.inserer_apres(Ma_Liste.tete,360)
# Ma_Liste.affichage_liste()
# Ma_Liste.inserer_apres_position(8,100)
# Ma_Liste.affichage_liste()
# Ma_Liste.supprimer_une_cellule(9)
# Ma_Liste.affichage_liste()
# Ma_Liste.supprimer_une_cellule(360)
# Ma_Liste.affichage_liste()
# print("Taille liste ",Ma_Liste.taille())
# """
# """
# Simple_list = []
# for i in range(1000000) :
#     Simple_list.append(i)
#
#
#
# deb1  = time.time()
# print(deb1)
# Simple_list.append(126)
# fin1 = time.time()
# print(fin1)
#
# print("Durée execution : ' ", fin1-deb1)
#
# """
```

```
213| #
214|
215|
216|
217|
218|
```