

```
# tptri.py
```

```
001 | from random import randint
002 |
003 | def liste_aleatoire(n,a,b):
004 |     L=[]
005 |     for i in range(n):
006 |         u=randint(a,b)
007 |         L=L+[u]
008 |     return(L)
009 |
010 | print(liste_aleatoire(100,0,10))
011 |
012 |
013 | def mini(L):
014 |     m=L[0]
015 |     for x in L:
016 |         if x<m:
017 |             m=x
018 |     return(m)
019 |
020 | def maxi(L):
021 |     m=L[0]
022 |     for x in L:
023 |         if x>m:
024 |             m=x
025 |     return(m)
026 |
027 |
028 | def second_mini(L):
029 |     m0=L[0]
030 |     m1=L[1]
031 |     for x in L:
032 |         if x<m0:
033 |             m0,m1=x,m0
034 |         elif x<m1:
035 |             m0,m1=m0,x
036 |     return(m1)
037 |
038 | def appartient(L,a):
039 |     for x in L:
040 |         if x==a:
041 |             return(True)
042 |     else:
043 |         return(False)
044 |
045 | def combien(L,a):
046 |     c=0
047 |     for x in L:
048 |         if x==a:
```

```

049|         c=c+1
050|     return(c) #Je change par rapport à l'énoncé, ici même
si a n'appartient pas à L on retourne un nombre, à savoir 0 au
lieu de retourner False.
051|
052|
053| def combien_index(L,a):
054|     L=[]
055|     for i in range(len(L)):
056|         if L[i]==a:
057|             L=L+[i]
058|     return(L)
059|
060| def supprimer_element(L,a):
061|     L2=[]
062|     for x in L:
063|         if L[i]!=a:
064|             L2=L2+L[i]
065|     return(L2)
066|
067| def supprimer_element_2(L,a):
068|     for x in L:
069|         if L[i]==a:
070|             L.remove(L[i])
071|     return(L)
072|
073| #autre solution pour second mini
074| def second_mini_2(L):
075|     m=mini(L)
076|     c=combien(L,m)
077|     if c>1:
078|         return(m)
079|     else:
080|         return(mini(supprimer_element(L,m)))
081|
082| def transform(S):
083|     ''' transforme une chaîne de caractère en liste, chaque
entrée de la liste étant une lettre'''
084|     L=[]
085|     for lettre in S:
086|         L=L+[lettre]
087|     return(L)
088|
089| def anagramme(S):
090|     a='' #chaîne de caractères vide.
091|     L=transform(S)
092|     for i in range(len(L)):
093|         u=randint(0,len(L)-1) #on choisit un indice au
hasard
094|         lettre=L[u] #on donne la lettre correspondante

```

```

095 |         a=a+lettre # on ajoute la lettre a l anagramme
096 |         L.remove(L[u]) # on enleve la lettre du mot de
base.
097 |     return(a)
098 | #test
099 | print(anagramme('CHAT'))
100 |
101 |
102 | def factoriel(n):
103 |     if n==0:
104 |         return(1)
105 |     else:
106 |         return(n*factoriel(n-1))
107 | #test
108 | print(factoriel(4))
109 |
110 | def lettre_distincte(S):
111 |     L=transform(S)
112 |     L_distincte=[]
113 |     for x in L:
114 |         if x in L_distincte:
115 |             L_distincte=L_distincte
116 |         else:
117 |             L_distincte=L_distincte+[x]
118 |
119 |     return(L_distincte)
120 |
121 | #test
122 | print(lettre_distincte('coucou'))
123 |
124 |
125 | def nombre_anagrammes(S):
126 |     L=transform(S)
127 |     N=factoriel(len(L))
128 |     L_distincte=lettre_distincte(S)
129 |     for lettre in L_distincte:
130 |         d=combien(L, lettre)
131 |         N=N/factoriel(d)
132 |
133 |     return(N)
134 |
135 | #test
136 | print(nombre_anagrammes('CHAT'))
137 | print(nombre_anagrammes('OII'))
138 |
139 |
140 | def nombre_anagrammes_2(S):
141 |     L=transform(S)
142 |     N=factoriel(len(L))
143 |     alphabet='abcdefghijklmnopqrstuvwxyz'

```

```

144 |
145 |     for lettre in alphabet:
146 |         d=combien(L, lettre)
147 |         N=N/factoriel(d)
148 |     return(N)
149 |
150 | #test
151 | print(nombre_anagrammes_2('chat'))
152 | print(nombre_anagrammes_2('oii'))
153 | #Attention minuscule et majuscule sont des lettres
différentes.
154 | #nombre_anagrammes_2('OII') ne va pas fonctionner
155 |
156 | def ordre_alphabetique(lettre1,lettre2):
157 |     alphabet='abcdefghijklmnopqrstuvxyz'
158 |     L_alphabet =transform(alphabet)
159 |
160 |     position_lettre1=0
161 |     position_lettre2=0
162 |     for i in range(len(L_alphabet)):
163 |         if L_alphabet[i]==lettre1:
164 |             position_lettre1=i
165 |         if L_alphabet[i]==lettre2:
166 |             position_lettre2=i
167 |
168 |     if position_lettre1<position_lettre2:
169 |         return(lettre1,lettre2)
170 |     else:
171 |         return(lettre2,lettre1)
172 |
173 | #test
174 | print(ordre_alphabetique('a','b'))
175 | print(ordre_alphabetique('h','d'))
176 |
177 |
178 |
179 |
180 |
181 |
182 |
183 |
184 |

```