

```

# Ch1-Prog.py

01| L=[6,87,-24,-8,94,74]
02|
03| def Recherche1(tab,val):
04|     presence=False # Initialisation de la variable contenant le résultat
05|     i=0 # Initialisation de la variable de boucle
06|     while i<len(tab) and not presence: # On parcourt tab tant que val n'est rencontré ou la fin atteinte
07|         if tab[i]==val:
08|             presence=True
09|             i=i+1
10|     return(presence)
11|
12|
13| #-----
14| #
15|
16| def Recherche2(tab,val):
17|     for i in range(0,len(tab)): # On parcourt a priori toute la séquence tab
18|         if tab[i]==val:
19|             return(True) # la présence de val implique qu'on renvoie True ce qui met fin à la boucle
20|     return(False)
21|
22|
23| #-----
24| #
25|
26| def Recherche3(tab,val):
27|     return(val in tab) # On renvoie le résultat (booléen) du test de présence de val dans tab
28|
29|
30| #-----
31| #
32|
33| def Recherche4(tab,val):
34|     indice=0 # Initialisation de la variable contenant le résultat
35|     while i<len(tab) and tab[indice]!=val: # On parcourt tab tant que val n'est rencontré ou la fin atteinte
36|         indice=indice+1
37|     return(indice) # Renvoie len(tab) si échec
38|
39|
40| #-----
41| #
42|
43| def Recherche5(tab,val):
44|     for indice in range(0,len(tab)): # On parcourt a priori toute la séquence tab
45|         if tab[indice]==val:
46|             return(indice) # la présence de val implique qu'on renvoie l'indice ce qui met fin à la boucle
47|     return(len(tab))
48|
49|
50| #-----
51| #
52|
53| def Recherche6(tab,val):
54|     for indice in range(1,len(tab)+1): # On parcourt a priori toute la séquence tab
55|         if tab[-indice]==val:
56|             return(len(tab)-indice) # la présence de val implique qu'on renvoie l'indice ce qui met fin à la boucle
57|     return(len(tab))
58|
59|
60| #-----
61| #
62|
63| def Recherche7(tab):
64|     max=tab[0] # Initialisation de la variable contenant le résultat
65|     for i in range(1,len(tab)): # On parcourt la totalité de la séquence à partir du deuxième élément
66|         if tab[i]>max:
67|             max=tab[i]
68|     return(max)
69|
70|
71| #-----
72| #
73|
74| def Recherche7bis(tab):
75|     max=tab[0] # Initialisation de la variable contenant le résultat
76|     indice=0
77|     for i in range(1,len(tab)): # On parcourt la totalité de la séquence à partir du deuxième élément
78|         if tab[i]>max:
79|             max=tab[i]
80|             indice=i
81|     return(max,indice)
82|

```