

```
# -*- coding: utf-8 -*-  
"""
```

```
Created on Wed Nov 6 10:09:51 2019
```

```
@author: dconduche  
"""
```

```
def dichoListe(L, x):  
    """Recherche la présence d'un élément dans une liste triée  
  
    L -- liste, supposée triée  
    x -- élément cherché  
    retourne le plus grand indice de l'élément ou -1 s'il n'est pas dedans  
    """  
    a = 0 # indice de début de liste  
    b = len(L) # indice de fin de liste  
    while b-a > 1:  
        m = (a+b)//2  
        if L[m] <= x:  
            a = m # l'indice de x est dans [[m, b[[  
        else:  
            b = m # l'indice de x est dans [[a, m[[ (ou x pas dans L)  
    if L[a] == x:  
        return a  
    else:  
        return -1
```

```
def dichoListeRec(L, x, i0=0):  
    """Recherche la présence d'un élément dans une liste triée  
  
    L -- liste, supposée triée  
    x -- élément cherché  
    i0 -- le décalage lié à la réindexation L[m:]  
    retourne le plus grand indice de l'élément ou -1 s'il n'est pas dedans  
    """  
    a = 0 # indice de début de liste  
    b = len(L) # indice de fin de liste  
    if b-a == 1:  
        if L[a] == x:  
            return i0  
        else:  
            return -1  
    m = (a+b)//2  
    if L[m] <= x:  
        return dichoListeRec(L[m:], x, i0+m)  
        # l'indice de x est dans [[m, b[[  
    else:  
        return dichoListeRec(L[:m], x, i0)  
        # l'indice de x est dans [[a, m[[ (ou x pas dans L)
```

```
# Test :  
L = list(range(-2, 11, 1))  
print(L)  
for x in [5.4, -2, 10, 3, -3, 11]:  
    i = dichoListeRec(L, x)  
    print('recherche de ' + str(x) + " dans la liste: " + str(i))
```