

```
def tri_insertion(L):
    n = len(L)

    # cas du tableau vide
    if n == 0:
        return L

    for j in range(1,n):
        e = L[j]
        i = j

        # A l'étape j, le sous-tableau L[0,j-1] est trié
        # et on insère L[j] dans ce sous-tableau en déterminant
        # le plus petit i tel que 0 <= i <= j et L[i-1] > L[j].
        while i > 0 and L[i-1] > e:
            i = i - 1

        # si i != j, on décale le sous tableau L[i,j-1] d'un cran
        # vers la droite et on place L[j] en position i
        if i != j:
            for k in range(j,i,-1):
                L[k] = L[k-1]
            L[i] = e
    return L

assert tri_insertion([2,5,-1,7,0,28]) == [-1, 0, 2, 5, 7, 28]
assert tri_insertion([10,9,8,7,6,5,4,3,2,1,0]) == [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```