

Nom :

Devoir 2 Nom :

Devoir 2

On propose ici l'élaboration d'un programme utilisant plusieurs fonctions et permettant ainsi le tri croissant d'une liste L d'entiers.

- 1) Créer une fonction `Parcours1(L)` qui :
 - parcourt une fois la liste L à l'aide d'une boucle `for`,
 - fait les échanges en permutant $L[i]$ et $L[i + 1]$ chaque fois que $L[i] > L[i + 1]$

Exemple : Si $L=[5,4,3,2,1]$ alors l'appel de `Parcours1(L)` modifie en place L . Ainsi L devient : $L=[4,3,2,1,5]$

- 2) La fonction précédente permet donc de faire remonter le nombre le plus grand en fin de liste. Proposer une fonction `tri1(L)` qui utilise plusieurs fois la fonction `parcours(L)` à l'aide d'une boucle `while`. Cette fonction retourne la liste L triée à la fin de son exécution

- 3) Créer une fonction `Parcours2(L,j)` qui :
 - parcourt une fois la liste $L[:j]$ à l'aide d'une boucle `for`,
 - fait les échanges en permutant $L[i]$ et $L[i + 1]$ chaque fois que $L[i] > L[i + 1]$

Exemple : Si $L=[5,4,3,2,1]$ alors l'appel de `Parcours2(L,3)` modifie en place L . Ainsi L devient : $L=[4,3,2,5,1]$

- 4) La fonction précédente permet donc de faire remonter le nombre le plus grand en fin de liste. Proposer une fonction `tri1(L)` qui utilise plusieurs fois la fonction `parcours(L)` à l'aide d'une boucle `while`. Cette fonction retourne la liste L triée à la fin de son exécution