

# Tri par insertion

Implémentez les exercices suivants en python. Les différents programmes doivent pouvoir traiter un tableau quel que soit son nombre de cases. On utilisera pour cela la fonction `len()` et des boucles `for`.

## Exercices (en python)

1/ Coder un programme qui, à partir d'un tableau, renvoie l'indice de son plus petit élément. Vous pourrez pour cela récupérer vos travaux du TP tri par sélection.

2/ Coder un programme qui à partir d'un tableau `[a, b, c, d, e]` renvoie `[e, a, b, c, d]`.

Vous pourrez utiliser les propriétés des listes en python, et leurs fonctions dédiées :

- l'opérateur `+` permet de fusionner deux listes ;
- `nomListe.insert(p, e)` insère l'élément `e` dans la liste `nomListe` à la position `p` ;
- `nomListe.append(e)` ajoute l'élément `e` à la fin de la liste `nomListe` ;
- `nomListe.pop()` renvoie l'élément final, et le supprime de la liste ;
- `nomListe.pop(p)` fait de même pour l'élément situé en position `p` ;
- `nomListe[i:j]` désigne la sous-liste comprenant les éléments de position `i` à `j`.

3/ Servez-vous des résultats des deux derniers exercices afin de coder un programme qui, à partir d'un tableau, place le plus petit de ses éléments au début, tout en conservant l'ordre de toutes ses autres cases.

Ainsi, le tableau `[a, b, c, d, e]` devient `[d, a, b, c, e]` dans le cas où `d` est le plus petit élément du tableau. Si son plus petit élément avait été `c`, alors le résultat aurait été `[c, a, b, d, e]`.

4/ Mettez en œuvre l'algorithme du tri par insertion. Le voilà décrit sur Wikipédia (11/2020) :

Dans l'algorithme, on parcourt le tableau à trier du début à la fin. Au moment où on considère le *i*-ème élément, les éléments qui le précèdent sont déjà triés. Pour faire l'analogie avec l'exemple du jeu de cartes, lorsqu'on est à la *i*-ème étape du parcours, le *i*-ème élément est la carte saisie, les éléments précédents sont la main triée et les éléments suivants correspondent aux cartes encore en désordre sur la table.

L'objectif d'une étape est d'insérer le *i*-ème élément à sa place parmi ceux qui précèdent. Il faut pour cela trouver où l'élément doit être inséré en le comparant aux autres, puis décaler les éléments afin de pouvoir effectuer l'insertion. En pratique, ces deux actions sont fréquemment effectuées en une passe, qui consiste à faire « remonter » l'élément au fur et à mesure jusqu'à rencontrer un élément plus petit.