

<b>TP 8 – Partie 2</b> Les tableaux à 1 dimension : tris de tableaux
---

<b>Exercice 1 : le tri par insertion</b>
--

Soit un tableau de noms `tNoms` de type `tabnoms`, avec le type `tabnoms` défini comme suit :

```
#define N 5
typedef char chaine20[21] ;
typedef chaine20 tabnoms[N] ;
```

1°) Écrire une procédure `affiche(tabnoms t, int n)` qui affiche à l'écran les  $n$  noms contenus dans  $t$ .

2°) Écrire une procédure `void maintientTrie (tabnoms t , int i)` qui rend le sous-tableau  $t[0..i]$  trié sachant que le sous-tableau  $t[0..i-1]$  est trié.

On fait donc « remonter » l'élément actuellement à l'indice  $i$ , à sa « vraie » place dans le sous-tableau  $t[0..i]$  (par exemple par permutation ; on peut aussi procéder par décalages cf `td7 ex3`).

3°) Écrire une procédure `void triParInsertion ( tabnoms t, int n)` qui trie le tableau  $t$  rempli de  $n$  valeurs, en faisant appel à la procédure `maintientTrie`.

Principe : on applique le raisonnement par récurrence

$i=0$  : le sous-tableau  $t[0..0]$  est trié !

$i$  quelconque : le sous-tableau  $t[0..i-1]$  étant trié, on passe à  $t[0..i]$  trié en appliquant la procédure `maintient_Trie`.

Quand  $i$  a atteint  $n-1$ , le tableau rempli jusqu'au rang  $n-1$  est trié.

4°) Écrire le programme principal qui

- initialise le tableau `tNoms` à la déclaration, par exemple :

```
tabnoms tNoms = {"toto", "zozo", "loto", "momo", "mimi"};
```

- affiche son contenu

- le trie

- affiche à nouveau son contenu

## Exercice 2 : le tri par sélection

Soit un tableau T de n entiers. Le principe du tri par sélection (ordre croissant) est de parcourir le tableau T par un indice I de sorte que la partie T[0..I-1] du tableau soit triée. On échange l'élément T[I] avec le minimum de la partie T[I..n-1] du tableau et la valeur de I est incrémentée.

Si T est un tableau d'entiers on peut avoir (avec I=3) :

2	6	7	14	10	11	8	19	15	9
---	---	---	----	----	----	---	----	----	---

Les éléments T[3] et T[6] sont échangés et I passe à 4.

**Question 1/** Appliquez le principe précédent pour trier, sur le papier, le tableau suivant :

4	1	5	9	10	11	7	3	8	6
---	---	---	---	----	----	---	---	---	---

**Question 2/** Comment doit se comporter le tri si plusieurs occurrences d'un même élément sont possibles ? Réfléchissez sur le tri du tableau suivant :

4	1	8	9	8	11	10	3	8	6
---	---	---	---	---	----	----	---	---	---

**Question 3/** En conclure la procédure :

```
void tri_selection(tab T) ;
```

qui trie le tableau T passé en paramètre.

**Question 4/** Que faut-il changer dans la procédure précédente pour obtenir un tri par sélection décroissant ?

## Exercice 3

Programmez et mettez au point les exercices du TD 8.