Algorithmique et structures de données
TP1 – Manipulation de tableaux, tris et récursivité

Lamia BENAMARA

Consigne :
Pour les séances TP, le langage de programmation utilisé sera le javascool/Robusta ou le java (blueJ).

Les premiers TP seront consacrés pour l'implémentation des algorithmes vus en TD.
Pensez toujours à soigner vos traces d'exécution.

Exercice 1 : Préambule – création et affichage de tableau

1. Ecrire une fonction qui prend en paramètre un entier n et un tableau d'entiers et qui demande à l'utilisateur de saisir les n nombres à stocker dans le tableau.
2. Ecrire une fonction qui prend en paramètre un tableau d'entiers qui sera rempli par des valeurs aléatoires comprises entre 1 et 100. Utiliser la fonction random.
3. Ecrire une fonction itérative (resp récursive) qui prend en paramètre un tableau et qui affiche son contenu.
4. Ecrire une méthode itérative (resp récursive) qui prend en paramètre un tableau et affiche son contenu dans le sens inverse (du dernier élément au premier élément).
5. Testez les fonctions précédentes dans le programme principal main.

Exercice 2 : traitements basiques sur un tableau

1. Ecrire une fonction itérative (resp récursive) « SommeTableau » qui prend en entrée un tableau « tab » d'entiers et qui retourne la somme de tous ses éléments.
2. Ecrire une fonction itérative (resp récursive) « EltPairTableau » qui prend en entrée un tableau « tab » d'entiers et qui retourne le nombre d'éléments pairs dans ce tableau.
3. Ecrire une fonction itérative (resp récursive) int OccurrenceTableau (int[] tab, int elt) permettant de retourner le nombre d'occurrences d'un élément "*elt*" dans un tableau d'entiers.
4. Ecrire une fonction itérative (resp r récursive) « InverserTableau» qui prend en paramètres un tableau tab et un indice « i » puis inverse l'ordre des éléments entre l'indice « i » et l'indice « N-i-1 » du tableau « tab » (pas de second tableau).
   Exemple : tab : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
   InverserTableau (tab, 0) => 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
   InverserTableau (tab, 3) => 1, 2, 3, 7, 6, 5, 4, 8, 9, 10

Exercice 3 : Implémentation des algorithmes de tri

Ecrire les fonctions correspondant aux algorithmes suivants (déjà vus en TD) :

1. Int Permutation(int tab[ ], int i, int j) : permute le contenu deux cases i et j du tableau

2. Int indice_minimum (int tab[ ], int i) : qui retourne l'indice de la valeur minimale du tableau en commençant la recherche à partir de l'indice i.
3. void tri_bulles(int tab[ ])
4. void tri_insertion(int tab[ ])
5. void tri_selection(int tab[ ]) : avec deux version : la première version doit se faire avec l'appel de la fonction indice_minimum, la deuxième sans.
6. Int recherche_dichotomique (int tab[ ], int e)
7. Testez vos fonctions dans le main, en affichant à chaque fois le tableau avant et après le tri.
8. Pensez toujours à sécuriser vos fonctions en ajoutant les tests nécessaires pour éviter des accès illégaux au niveau du tableau.