

Travaux Dirigés de Programmation II
[TD n°1 : Rappels & Notion d'Enregistrement]

Objectifs : - Rappeler les notions de tableaux, de pointeurs, de fonctions et de récursivité ;
- Introduire la notion d'enregistrement (structure en C).

Exercice 1

- a)- Ecrire en langage C une fonction itérative, **nbre_occurrences**, qui détermine le nombre de fois où un élément noté **elt** apparaît dans un tableau **T** donné contenant de **N** entiers.
- b)- Refaire la question a)- en utilisant une fonction récursive, **nbre_occ_rec**.
- c)- Ecrire une fonction principale, **main**, qui teste ces deux fonctions.

Exercice 2

Soit **T** un tableau d'entiers contenant **N** éléments. On suppose qu'il n'est pas trié. Définir une fonction de type **void**, **elt_plus_frequent**, qui détermine l'élément qui apparaît le plus de fois dans le tableau **T**, ainsi que son nombre d'occurrences. Si plusieurs éléments répondent au problème, alors prendre un seul. On notera qu'il n'est pas permis d'utiliser d'autres tableaux.

Exercice 3

On s'intéresse à définir une fonction effectuant la recherche dichotomique d'un élément noté **elt** dans un tableau **T** trié de **N** nombres entiers. Cette fonction retournera la position de l'élément **elt** dans le tableau **T**, sinon la valeur **-1**.

- a)- Donner une version itérative, **rech_dicho_iter**, de la fonction de recherche dichotomique.
- b)- Donner une version récursive, **rech_dicho_rec**.

Exercice 4

Définir la fonction, **tri_selection**, qui effectue le tri par ordre croissant d'un tableau **T** de **N** entiers par la méthode de sélection. On rappelle que le principe du tri par sélection consiste à échanger le premier et le plus petit élément de la partie non triée du tableau. Pour cela, on commencera par définir les deux fonctions : **echanger** et **position_minimum**.

Exercice 5

On caractérise un étudiant par son matricule (*entier*), son nom (*chaîne de 20 caractères au plus*), son prénom (*chaîne de 20 caractères au plus*), sa date de naissance (*3 entiers désignant le jour, le mois et l'année*) et sa moyenne générale (*réel*).

- a)- Définir en langage C le type **Etudiant**, représentant un étudiant. On notera que la date de naissance pourra être stockée dans une structure à part. (*indication : utiliser struct et typedef*).

On considère la structure **Tab_Etud** suivante permettant de gérer 1000 étudiants.

```
typedef struct tab_etud {
    int nb_etud; /* nombre d'étudiants */
    Etudiant etudiants[1000]; /* tableau d'étudiants */
} Tab_Etud;
```

- b)- Ecrire une fonction, **saisie_etudiant**, de saisie des informations d'un étudiant. Il est à noter qu'on ne se préoccupera pas de la validité de la date entrée.
- c)- Ecrire une fonction, **affiche_etudiant**, d'affichage des informations concernant un étudiant.
- d)- Ecrire une fonction, **ajout_etudiant**, qui permet d'ajouter un étudiant dans un tableau d'étudiants
- e)- Ecrire une fonction, **tri_nom**, qui permet de classer les étudiants par ordre alphabétique des noms.