Université Mohammed V-Agdal
Faculté des Sciences de Rabat
Département d'Informatique

Année Universitaire 2013/2014
MODULE M14 (SMI-S4)

Travaux Dirigés de Structures de Données
[TD n°1 : Rappels & Complexité Algorithmique]

Objectifs : *- Rappeler les notions de tableaux, de pointeurs, d'enregistrements et de récursivité ;*
- Initier au calcul de la complexité algorithmique.

Exercice 1

On caractérise un étudiant par son matricule (*entier*), son nom (*chaîne de 20 caractères au plus.*), son prénom (*chaîne de20 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.

Exercice 2

a)- Soit l'algorithme itératif **`a_iter`** suivant :

```
Algorithme a_iter(n : entier) : entier
Entrée : un entier
Sortie : un entier ?
Début
Variables locales : i, r : entier
r ← 2
Pour i de 1 à n faire
  r ← r * r
Fpour
retourner r
Fin
```

Que calcule cet algorithme ? Proposer une version récursive **`a_rec`** de cet algorithme.

b)- Ecrire une fonction récursive, **`moy_rec`**, qui calcule la moyenne de n nombres réels (n non nul).

c)- Ecrire une fonction récursive, **`recherche_dicho`**, qui effectue la recherche dichotomique dans un tableau trié de n nombres entiers.

Exercice 3

Analyser, en utilisant la notation Grand-O, les deux parties de code C suivantes :

Partie de code 1

```
x=0;                        // 1
for (i=1; i<=n; i++)        // 2
   x++;                     // 3
for (j=1; j<=n; j=j*2)      // 4
   x++;                     // 5
```

Partie de code 2

```
x=0;                            // 1
for (i=1; i<=n/2; i++)          // 2
   for (j=1; j<=n; j=j*2)       // 3
      x++;                      // 4
```

Exercice 4

On se donne un tableau T de n nombres entiers (n >= 1). Ecrire une fonction, **`moy_prefixe`**, qui construit, à partir de T, le tableau MP de même taille n, tel que MP[i] est la moyenne de T[0], T[1], ...,T[i] pour i = 0, 1, …, n-1. Evaluer sa complexité.

Exercice 5

Ecrire une fonction itérative, **`affiche_chiffres`**, qui affiche les chiffres composant un nombre entier positif n. Evaluer sa complexité.