



Travaux Dirigés d'informatique n°2  
Algorithmique  
Rappels sur la récursivité et table de hachage.

**Exercice 1** On considère la fonction récursive en langage C suivante:

```
void F(int n, int base){
    if (n>0){
        F(n/base, base);
        printf("%d ", n%base);
    }
}
```

1. Déroulez l'algorithme sur un exemple.
2. Que se passe-t'il si nous inversons la troisième ligne et quatrième ligne.
3. Quelle est la complexité de la fonction  $F(n, b)$ . On supposera  $b > 0$ .

**Exercice 2** 1. Rappelez ce qu'est une table de hachage ouvert par liste chaînée et une table de hachage direct.

2. On considère la phrase suivante

*Maitre corbeau sur un arbre perche tenait en son bec un fromage*

Construisez la table de hachage de taille  $N$  contenant le lexique des mots (avec multiplicité) de la phrase ci-dessus. On utilisera pour ce faire les fonctions de hachages suivantes

- $$h_1(w) = \sum_{0 \leq i \leq l(w)-1} e(w_i) \pmod{N}$$
- $$h_2(w) = \sum_{0 \leq i \leq l(w)-1} (i+1) e(w_i) \pmod{N}$$

pour  $N = 5$ , et  $N = 50$  où  $l(w)$  est la longueur du mot  $w$  et  $e(w)$  est la position de la lettre  $w_i$  dans l'alphabet (ordonné dans l'ordre lexicographique). C'est à dire  $e(a) = 1$ ,  $e(b) = 2, \dots$ . Testez le hachage ouvert par liste chaînée et le hachage fermé linéaire.

3. Que pensez vous des fonctions pour  $N = 10000$ .