

# Travaux Dirigés d'Algorithmique n°3

## Cours d'Informatique de Première Année

— Licence 1.2 —

---

### Récurtivité

Ce TD porte sur des algorithmes récurtifs.

---

► **Exercice 1.** Exprimer récurtivement les fonctions suivantes et donner le pseudo-code correspondant.

1. la fonction factorielle
2. la fonction puissance ( $a^n$  avec  $0 \leq n$ )
3. la fonction de calcul du pgcd
4. la fonction de recherche dichotomique du maximum d'un tableau non trié et non vide
5. la fonction vérifiant si une chaîne de caractères dont la taille est donnée en paramètre est un palindrome

► **Exercice 2.**

1. Écrire une fonction `SommeChiffre` prenant en argument un entier positif  $n$  et qui calcule récurtivement la somme des chiffres qui le forment. Par exemple si  $n = 3281$  alors la fonction doit retourner  $14 = 3 + 2 + 8 + 1$ .
2. La propriété suivante est bien connu : le reste de la division euclidienne d'un entier par 3 est égale au reste de la division euclidienne de la somme de ses chiffres par 3. Autrement dit : Si  $n$  s'écrit avec  $k$  chiffres en base 10 et que  $c_i$  est le  $i$ -ème chiffre en partant de la droite. Alors

$$\sum_{i=1}^{i=k} c_i \equiv n \pmod{3}$$

On remarquera que la somme des chiffres d'un entier supérieur à 10 est toujours strictement plus petite que cet entier. Utiliser cette remarque pour écrire une fonction récurtitive qui retourne la division euclidienne d'un entier par 3 (sans utiliser l'opération modulo, évidemment).

3. Effectuer le même travail pour le reste de la division euclidienne par 11 en remarquant que si  $c_i$  est le  $i$ -ième chiffre de  $n$  en partant de la droite alors :

$$\sum_{i=1}^{i=k} (-1)^{1+i} c_i \equiv n \pmod{11}$$

Vous pouvez écrire et utiliser une nouvelle fonction entier `SommeAlternee(entier n)` qui calcul la somme des chiffres d'un entier passé en paramètre en alternant les signes. Par exemple `SommeAlternee(17328) = 8 - 2 + 3 - 7 + 1 = 3`  
Remarquer que  $8 - 2 + 3 - 7 + 1 = 8 - (2 - (3 - (7 - (1))))$

► **Exercice 3. Somme d'entiers (problème du sac à dos)**

On dispose d'un ensemble  $E$  d'entiers strictement positifs et d'un entier positif  $S$ . Le problème consiste à déterminer s'il existe un sous-ensemble de  $E$  tel que la somme de ses éléments soit égale à  $S$ .

Par exemple, si  $E = \{3, 4, 5, 12, 14, 26\}$  et  $S = 21$ , la réponse est oui et le sous-ensemble est  $\{4, 5, 12\}$ .

Montrer sur l'exemple précédent qu'une méthode gloutonne ne fournit pas la solution.

Décrire récursivement une recherche de la solution du problème. On considère que la somme des éléments de l'ensemble vide est 0.

Écrire une fonction `sacADos(entier element,entier sac[],entier taille)`, l'ensemble étant représenté sous forme d'un tableau de `taille` éléments. La fonction doit retourner `VRAI` et afficher les éléments du sous-ensemble si une solution existe.

Pour un ensemble de taille  $n$  quel est le nombre maximal d'appels récursifs que la fonction peut effectuer avant de résoudre le problème? Résoudre le problème dans notre cas signifie trouver une solution ou s'assurer qu'il n'en existe pas.