

# Travaux Pratiques n°6

## Exercice 1

Implémenter les fonctions dans les exercices 4 et 5 du TD6 et les tester.

## Exercice 2

Implémenter les fonctions dans les exercices 6,7 et 8 du TD6 et les tester.

## Exercice 3

Écrire une fonction `matrice_mult` qui prend comme argument 3 matrices NxN  $A$ ,  $B$  et  $C$  et qui calcule le produit  $C = A \times B$ .

## Exercice 4

On veut écrire un algorithme permettant de convertir le nombre en chiffre romain vers des nombres usuels.

L'écriture des nombres par les romain utilise les symboles I=1, V=5, X=10, L=50, C=100, D=500, M=1000. Par exemple :

$$(I, II, III, IV, V, VI, VII, VIII, IX, X) = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

$$(XLI, XLII, XLIII, XLIV) = (41, 42, 43, 44)$$

Un nombre en chiffres romain ne peut jamais avoir quatre symboles consécutif identiques. On suppose que le nombre à convertir est stocké dans une chaîne de caractère.

1. Écrire une fonction `valeur` qui prend un caractère comme paramètre et rend sa valeur décimale, si le caractère n'est pas un chiffre romain, la fonction rend 0.
2. Écrire une fonction `plus_grand` qui prend 2 caractères représentant des chiffres romains  $a$  et  $b$  comme paramètres et rend 0 si  $a = b$ , -1 si  $a < b$  et 1 si  $a > b$ .
3. Écrire une fonction qui prend une chaîne de caractère représentant nombre romain et qui rend sa valeur en décimale. La fonction rend 0 si la chaîne ne représente pas un nombre romain.

*Suggestion de l'algorithme : On parcourt la chaîne de caractère  $c$  représentant le nombre romain et on initialise  $val = 0$ . On met  $n =$  la longueur de  $c$  et on maintient une variable  $temp$  et un compteur. A l'itération  $i$  ( $i = 0$  à  $n - 1$ ) on ajoute à  $temp$  la valeur de  $c[i]$ .*

(a) Si  $c[i + 1] < c[i]$  ou  $i = n - 1$  (le dernier caractère) alors on ajoute à  $val$  la valeur de  $temp$ .

(b) si  $c[i + 1] = c[i]$  alors on incrémente compteur de 1, si ce compteur  $> 2$  (expliquez pourquoi ?), on sort de la boucle et termine la fonction en rendant 0.

(c) si  $c[i + 1] > c[i]$  alors  $temp = -temp$ .

Sauf pour le cas (b), dans tous les cas, à la fin de chaque itération, on reinitialise compteur à 0. Dans le cas 1, on reinitialise temp à 0.

4. L'algorithme suggéré précédemment accepte par exemple IXXX comme un nombre correct alors ce n'est pas le cas (le nombre correct est XXIX). Modifier la fonction précédente pour prendre en compte ce cas.