

Vous devez mettre en ligne chez votre hébergeur les diverses pages WEB réalisées. Un accès individuel à chaque page devra se faire via un lien actif placé sur la page d'accueil index.htm.

Pour chacun des exercices vous devez codifier en PHP l'algorithme demandé. Les saisies doivent **obligatoirement** faire l'objet d'un formulaire interne avec, lorsque c'est possible, un maximum de tests de validité.

Exercice 1 : nombre d'Armstrong

Un nombre d'**Armstrong** est un entier pouvant se représenter par la somme des cubes de ses chiffres.

$$\text{Exemple : } 153 = 1^3 + 5^3 + 3^3$$

Etablir l'algorithme complet (i.e. arbre programmatique et environnement) permettant de déterminer si un nombre entier donné est un nombre d'**Armstrong** (*Remarque* : vous pouvez décomposer le nombre pour extraire ses chiffres en utilisant des divisions successives par 10 et en "travaillant" les quotients et les restes). Il vous est **interdit** d'utiliser les tableaux.

Exercice 2 : palindrome numérique

Un nombre entier peut être considéré comme un **palindrome**¹ **numérique** s'il garde la même valeur en inversant les chiffres le constituant.

$$\text{Exemple : } 12321, 101, 4$$

Etablir l'algorithme complet permettant de déterminer si un nombre entier est un **palindrome numérique** ou non. Il vous est **interdit** d'utiliser les tableaux (cf. exercice 1)

¹ Un palindrome est un mot (ou une phrase) qui peut être lu indifféremment de gauche à droite ou de droite à gauche (*Exemple* : "kayak", "radar", "Un roc cornu", "Esopé reste ici et se repose", "Karine alla en Irak", ...).

Exercice 3 : multiplication par additions successives

Le produit de deux nombres entiers positifs ou nuls peut être obtenu par additions successives du multiplicande autant de fois que nécessaire.

$$a \times b = \underbrace{a + a + a + \dots + a}_{b \text{ fois}}$$

Dans un souci d'optimisation, le produit étant commutatif, il est efficace de permuter le multiplicande et le multiplicateur pour toujours obtenir un nombre minimal d'additions.

Etablir l'algorithme complet permettant la multiplication de 2 entiers par additions successives.

Exercice 4 : exponentiation indienne

L'exponentiation indienne permet de calculer x^y , x et y étant deux nombres entiers positifs ou nuls, en ne disposant que de l'élévation au carré, que de la division par deux et que de la multiplication de deux nombres avec un minimum d'opérations à effectuer.

1. Dans le cas où y est pair les facteurs x sont regroupés deux par deux pour minimiser le nombre de multiplications.

En effet	$x \times x \times x \times x \times x \times x \times x \times x$	$y = 8$
$\Leftrightarrow$	$x^2 \times x^2 \times x^2 \times x^2$	$y = 4$
$\Leftrightarrow$	$x^4 \times x^4$	$y = 2$
$\Leftrightarrow$	x^8	$y = 1$

il n'y a que trois multiplications à effectuer :

$$\begin{array}{l} x \times x \\ x^2 \times x^2 \\ x^4 \times x^4 \end{array}$$

alors que le calcul classique nécessiterait sept multiplications.

2. Dans le cas où y est impair on se ramène au cas précédent en extrayant un facteur :

En effet	$x \times x \times x \times x \times x \times x$	$y = 5 \text{ à } y = 4$
$\Leftrightarrow$	$x^2 \times x^2$	$y = 2$
$\Leftrightarrow$	x^4	
$\Leftrightarrow$	x^5	$y = 1$

il y a trois multiplications au lieu de quatre avec le calcul habituel. L'intérêt de cette méthode est évident lorsqu'il y a une très grande valeur de y .

Etablir l'algorithme complet permettant l'exponentiation indienne.

Exercice 5 : méthode de Newton

La méthode de Newton pour calculer une approximation de la racine carrée d'un nombre x peut être décrite par :

- Si y est une approximation de $\sqrt{x}$ alors $(y + x/y)/2$ en est une meilleure.
- Calculer les approximations successives jusqu'à ce que le résultat soit assez bon. On suppose que y est une assez bonne estimation de $\sqrt{x}$ si $|x - y^2| < \epsilon$.

Etablir l'algorithme complet permettant l'extraction de la racine carrée d'un nombre x à ϵ près par la méthode de Newton.

Exercice 6 : compression d'un vecteur

Soit un vecteur $U1$ de composantes $u_1, u_2, \dots, u_n$ non nulles et L un vecteur de même longueur dont les composantes $l_1, l_2, \dots, l_n$ sont 0 ou 1. La compression de $U1$ par L est le vecteur $U2$ dont les composantes sont, dans l'ordre, celles de $U1$ pour lesquelles la composante de L vaut 1.

Exemple

$$\begin{aligned} U1 &= \{1, 2, 3, 4, 5, 6, 7\} \\ L &= \{0, 1, 1, 0, 1, 0, 1\} \\ U2 &= \{2, 3, 5, 7, 0, 0, 0\} \end{aligned}$$

Etablir l'algorithme complet d'une telle compression de vecteur.

Exercice 7 : recherche de la première plus longue monotonie

Soit un vecteur T de N (≤ 25) éléments réels.

Etablir l'algorithme complet déterminant la longueur de la première plus longue séquence de nombres rangés par ordre croissant et le rang de son premier élément.