

Instructions conditionnelles, fonctions récursives

Exercice 1 *L'exécution d'une division, lorsque le diviseur vaut 0, provoque un arrêt du programme (avec un message plus ou moins incompréhensible).*

Donner la sémantique (paramètres, retour, effets de bord) de chacune des fonctions C suivantes :

```
int quot(int a,int b) {
    return a/b;
}

int quasiquot(int a,int b) {
    if(b == 0) return 0;
    else return a/b;
}

int aliquot(int a, int b) {
    if(b == 0) {
        printf("erreur : division par zéro\n");
        return 0;
    }
    return a/b;
}

int tartiquot(int a, int b) {
    if(b == 0) printf("erreur : division par zéro\n");
    return a/b;
}
```

Exercice 2 *Écrire une fonction maximum qui retourne le maximum de trois entiers x, y, z, donnés.*

Exercice 3 *Écrire une fonction `pastropnul` qui, étant donnés trois entiers x, y, z , retourne 1 si au plus l'un de ces trois entiers est nul, et 0 sinon.*

Exercice 4 *On considère la fonction `C` suivante :*

```
int q(int a,int b) {
    if(a < b) return 0;
    else return 1 + q(a-b,b);
}
```

1. *Faire tourner l'algorithme pour $a = 19$ et $b = 7$.*
2. *D'une manière générale, que calcule la fonction `q` dans le cas où a et b sont des entiers positifs ?*
3. *Que se passe-t-il si a ou b est négatif ou nul ?*

Exercice 5 *Écrire une fonction récursive `somme_cubes` qui, pour un entier n donné, calcule la somme des n premiers cubes.*

Exercice 6 *On considère la fonction `C` suivante :*

```
int f(int a) {
    int r;
    if(a <= 0) r = 0;
    else r = a*f(a-1);
    return a+r;
}
```

1. *Simplifier le code de `f` de façon à supprimer l'utilisation de la variable locale `r`.*
2. *Évaluer l'expression `f(5)`.*

Exercice 7 *Écrire une fonction récursive `somme_chiffres` qui, pour un entier n donné, calcule la somme de ses chiffres en base décimale.*