

Exercice : calcul de factorielle

Exercice n°32 (pages 81 et 251) de l'ouvrage *C++ par la pratique*.

- Pour la fonction `factorielleIterative()`, il s'agit d'une simple boucle `for`. Il faut simplement veiller aux bonnes conditions initiales :

```
/* -----  
 * Calcule de façon itérative la factorielle d'un nombre entier positif.  
 * Entrée : le nombre n dont on veut calculer la factorielle  
 * Sortie : n!  
 * ----- */  
unsigned int factorielleIterative(unsigned int n)  
{  
    unsigned int fact(1);  
    for (unsigned int i(2); i <= n; ++i)  
        fact = fact * i;  
    return fact;  
}
```

Je vous conseille de toujours **commenter vos programmes**, et en particulier de dire ce que fait chaque fonction, quels sont ses arguments et à quoi correspond la valeur de retour.

- Pour la fonction `factorielleRecursive()` :

```
/* -----  
 * Calcule de façon récursive la factorielle d'un nombre entier positif.  
 * Entrée : le nombre n dont on veut calculer la factorielle  
 * Sortie : n!  
 * ----- */  
unsigned int factorielleRecursive(unsigned int n)  
{  
    if (n == 0)  
        return 1;  
    else  
        return (n * factorielleRecursive(n-1));  
}
```

Cette fonction récursive est construite sur le schéma vu en cours :

- a. Détermination de la condition d'arrêt :

```
if (n == 0) return 1;
```

On pourrait même optimiser un peu plus et écrire :

```
if (n <= 1) return 1;
```

- b. Écriture de la relation de récurrence :

```
n * factorielleRecursive(n-1)
```

- Il faut maintenant écrire le corps principal du programme (fonction `main()`).

La base est assez simple :

- a. Déclarer une variable entière :

```
unsigned int n;
```

- b. Demander sa valeur. Tant qu'à faire pourquoi ne pas le faire lors de l'initialisation de cette variable ? :

```
unsigned int n(demander_nombre(0, 12));
```

- c. Appeler successivement les deux méthodes et afficher les résultats :

```
cout << "Méthode itérative : " << endl;  
cout << "      " << n << "! = " << factorielleIterative(n) << endl;  
cout << "Méthode récursive : " << endl;
```

```
cout << " " << n << "! = " << factorielleRecursive(n) << endl;
```

- La boucle demandant à l'utilisateur s'il souhaite recommencer se fait de façon usuelle.

On a donc finalement le programme suivant :

```
#include <iostream>
#include <limits>
#include <string>
using namespace std;

/* -----
 * Calcule de façon itérative la factorielle d'un nombre unsigned int positif.
 * Entrée : le nombre n dont on veut calculer la factorielle
 * Sortie : n!
 * ----- */
unsigned int factorielleIterative(unsigned int n)
{
    unsigned int fact(1);
    for (unsigned int i(2); i <= n; ++i)
        fact *= i;
    return fact;
}

/* -----
 * Calcule de façon récursive la factorielle d'un nombre unsigned int positif.
 * Entrée : le nombre n dont on veut calculer la factorielle
 * Sortie : n!
 * ----- */
unsigned int factorielleRecursive(unsigned int n)
{
    if (n == 0)
        return 1;
    else
        return (n * factorielleRecursive(n-1));
}

/* -----
 * fonction demandant à l'utilisateur un nombre compris
 * dans un intervalle [a, b], ou supérieur ou égal à a
 * si b < a.
 * Entrée : les deux nombres a et b définissant l'intervalle
 * Sortie : le nombre saisi par l'utilisateur
 * ----- */
unsigned int demander_nombre(unsigned int a, unsigned int b)
{
    unsigned int res;

    do {
        cout << "Entrez un nombre entier ";
        if (a >= b)
            cout << "supérieur ou égal à " << a;
        else
            cout << "compris entre " << a << " et " << b;
        cout << " : ";
        cin >> res;
        if (cin.fail()) {
            cout << "Je vous ai demandé d'entrer un nombre, "
                << "pas du charabia !" << endl;
            // remet cin dans un état lisible
            cin.clear();
            // "jette" toute la ligne
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            res = a-1;
        }
    } while ((res < a) or ((a < b) and (res > b)));

    return res;
}
```

```
}  
  
// -----  
int main()  
{  
    char rep;  
  
    do {  
        unsigned int n(demander_nombre(0, 12));  
        cout << "Méthode itérative :" << endl;  
        cout << "      " << n << "! = " << factorielleIterative(n)  
            << endl;  
        cout << "Méthode récursive :" << endl;  
        cout << "      " << n << "! = " << factorielleRecursive(n)  
            << endl;  
  
        do {  
            cout << "Voulez-vous recommencer [o/n] ? ";  
            cin >> rep;  
        } while ((rep != 'o') and (rep != 'n'));  
    } while (rep == 'o');  
  
    return 0;  
}
```