

Calcul de PGDC

Exercice n°38 (pages 90 et 272) de l'ouvrage *C++ par la pratique*.

Sans surprise le prototype de la fonction pour calculer le PGDC sera :

```
int pgdc(int a, int b);
```

La seule difficulté de cet exercice réside dans la bonne réalisation des calcul de ces suites x , y , u et v qui s'appellent les unes les autres.

Il faut faire particulièrement attention à l'ordre des calcul pour être sûr(e) d'utiliser les bonnes valeurs.

Avec l'aide de l'énoncé, on définit et initialise

```
int prev_u(1), prev_v(0), x(a), y(b), u(0), v(1),
    new_u, new_v, q, r;
```

Le calcul peut alors se faire sans piège :

```
q = x/y;
r = x%y;
x = y;
y = r;
new_u = prev_u - q * u;
new_v = prev_v - q * v;
```

Mais il ne faut pas oublier de remettre à jour les valeurs servant à mémoriser :

```
prev_u = u;
u = new_u;
prev_v = v;
v = new_v;
```

On a donc finalement le programme suivant :

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

int pgdc(int a, int b);
int demander_nombre(int min, int max);

// -----
int main()
{
    int a(demander_nombre(1, 0));
    int b(demander_nombre(1, 0));
    cout << "PGCD(" << a << ", " << b << ")=" << pgdc(a,b) << endl;
    return 0;
}

// -----
int pgdc(int a, int b)
{
    int prev_u(1), prev_v(0), x(a), y(b), u(0), v(1);
    int new_u, new_v, q, r;

    /* Juste pour info. Normalement une telle fonction NE doit RIEN
     * afficher (« effet de bord ») !
     */
    cout << "Calcul du PGDC de " << a << " et " << b << endl;
    cout << endl;
    cout << "      x          y          u          v" << endl;
```

```

while (y != 0) {
    q = x/y;
    r= x%y;
    x = y;
    y = r;
    new_u = prev_u - u * q;
    prev_u = u;
    u = new_u;
    new_v = prev_v - v * q;
    prev_v = v;
    v = new_v;

    /* Juste pour info. Normalement une telle fonction NE doit RIEN
     * afficher (« effet de bord ») !
     * L'affichage est ici approximatif. Pour faire mieux il faudrait
     * utiliser setw (présenté plus tard en cours)
     */
    cout << "      " << x
         << "      " << y
         << "      " << u
         << "      " << v
         << endl;
}

return x;
}

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

    do {
        cout << "Entrez un nombre entier ";
        if (a >= b) cout << "superieur ou egal a " << a;
        else cout << "compris entre " << a << " et " << b;
        cout << " : ";
        cin >> res;
    } while ((res < a) or ((a < b) and (res > b)));
    return res;
}

```