MOOC Init. Prog. C++
Correction des exercices supplémentaires semaine 3

Les corrigés proposés correspondent à l'ordre des apprentissages : chaque corrigé correspond à la solution à laquelle vous pourriez aboutir au moyen des connaissances acquises jusqu'à la semaine correspondante.

Exercice 2 : histoire de prêt

> Cet exercice correspond à l'exercice n°8 (pages 22 et 204) de l'ouvrage *C++ par la pratique* (3e édition, PPUR).

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

int main()
{
    double S0(0.0);
    do {
        cout << "Somme prêtée                    (S0 > 0) : ";
        cin >> S0;
    } while (S0 <= 0.0);

    double r(0.0);
    do {
        cout << "Montant fixe remboursé chaque mois (r > 0) : ";
        cin >> r;
    } while (r <= 0.0);

    double ir(0.0);
    do {
        cout << "Taux d'intérêt en % (0 < tx < 100) : ";
        cin >> ir;
    } while ( (ir <= 0.0) or (ir >= 100.0) );
    ir /= 100.00;

    double cumul(0.0); // somme des intérêts (cumulés)
    double S(S0);      // somme restant à rembourser
    int nbr(0);        // nombre de remboursements

    while (S > 0.0) {
        ++nbr;
        cumul = cumul + ir * S;
        S = S - r;
        cout << nbr << ": S=" << S << ", cumul=" << cumul << endl;
    }
    cout << "Somme des intérêts encaissés : " << cumul
```

```
          << "  (sur " << nbr << " mois)." << endl;
    return 0;
}
```

Remarque : on aurait aussi pu écrire la boucle à l'aide d'une itération `for`, par exemple :

```
for (double S(S0); S > 0.0; S -= r) {
  ++nbr;
  cumul += ir * S;
}
```

Exercice 3 : Suite et série

a)

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

int main()
{
  double U(1.0);

  cout << "U0 = " << U << endl;
  for (int n(1); n <= 10; ++n) {
    U = U / n; // ou aussi : U /= n;
    cout << "U" << n << " = " << U << endl;
  }

  return 0;
}
```

b)

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

int main()
{
  double U(1.0);
  double V(U); // V0 = U0

  for (int n(1); n <= 10; ++n) {
    U = U / n; // ou aussi : U /= n;
    V = V + U; // ou aussi : V += U;
    cout << "U" << n << " = " << U
         << "        V" << n << " = " << V
         << endl;
  }

  return 0;
}
```

Exercice 4 : Figures en mode texte

L'affichage d'une ligne se fait à l'aide d'une simple boucle `for`. Pour afficher plusieurs lignes, il faudra imbriquer 2 boucles `for`.

Pour le triangle, la longueur d'une ligne est fonction du numéro de ligne. Donc, la condition d'arrêt de la boucle écrivant une ligne dépendra du compteur de la boucle parent (`j`).

Pour la pyramide inversée, il suffit de rajouter un nombre variable d'espaces avant les nombres.

```
#include <iostream>

using namespace std;

int main()
{
  // 1.
  for (int i(1); i <= 9; ++i) {
    cout << i;
  }
  cout << endl << endl;

  // 2.
  for (int j(1); j <= 9; ++j) {
    for (int i(1); i <= 9; ++i) {
      cout << i;
    }
    cout << endl;
  }
  cout << endl;

  // 3.
  for (int j(1); j <= 9; ++j) {
    for (int i(1); i <= j; ++i) { // la boucle va de 1 a j !!
      cout << i;
    }
    cout << endl;
  }
  cout << endl;

  // 4.
  for (int j(1); j <= 9; ++j) {
    for (int i(1); i <= (9-j); ++i) {
      cout << " ";
    }
    for (int i(1); i <= j; ++i) {
      cout << i;
    }
    cout << endl;
  }
  cout << endl;

  return 0;
}
```

Exercice 5 : Triangle

La difficulté de cet exercice consiste essentiellement à trouver la relation entre le numéro de ligne et le nombre d'étoiles et d'espaces. Soit n le nombre total de lignes, et i le numéro de ligne, commençant à 0. Le nombre d'espaces est donné par $s = n - i$, et le nombre d'étoiles vaut $e = 2 i + 1$.

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

int main()
{
  cout << "Entrez le nombre de lignes du triangle : ";
  int n;
  cin >> n;

  for (int i(0); i < n; ++i) {
    // ecrit les espaces avant le triangle
    for (int j(1); j < n - i; ++j) {
      cout << " ";
    }

    // ecrit les etoiles du triangle
    for (int j(0); j < 2 * i + 1; ++j) {
      cout << "*";
    }

    // retour a la ligne
    cout << endl;
  }

  return 0;
}
```

Exercice 6 : calcul de PGDC

> Cet exercice correspond à l'exercice n°38 (pages 90 et 272)
> de l'ouvrage *C++ par la pratique* (3e édition, PPUR).

Le début est assez classique et ressemble beaucoup aux programmes `somme.cc` et `factorielle.cc` fait la semaine passée.

La nouveauté réside uniquement dans la fonction `pgdc`.

Sans surprise son prototype 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'appelent 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;
```

```
/* Voici une version simplifiée par rapport à celle du livre
 * puisque nous n'avons pas encore vu les fonctions
 */
#include <iostream>
#include <cmath>
using namespace std;
```

```cpp
int main()
{
  int a(0);
  cout << "Entrez un nombre entier supérieur ou égal à 1 : ";
  cin >> a;

  int b(0);
  cout << "Entrez un nombre entier supérieur ou égal à 1 : ";
  cin >> b;

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

  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;

    // affichage approximatif. Pour faire mieux il faudrait utiliser setw (non
    cout << "    "      << x
         << "       "   << y
         << "         " << u
         << "     "     << v
         << endl;
  }

  cout << "PGDC(" << a << "," << b << ")=" << x << endl;

  return 0;
```