

**Exercice 1.** *Euclide : version récursive terminale.*

Le plus grand commun diviseur (**pgcd**) de deux entiers  $a$  et  $b$  peut être calculé via l'algorithme d'Euclide. Celui-ci est caractérisé par l'équation suivante.

$$\text{pgcd}(a, b) = \begin{cases} a & \text{si } b = 0 \\ \text{pgcd}(b, a \bmod b) & \text{sinon} \end{cases}$$

1. Donnez une implémentation récursive terminale de l'algorithme d'Euclide.
2. Donnez une implémentation itérative de l'algorithme d'Euclide.

**Correction :**

```
1 public static int pgcd(int a, int b) {
2     if (b == 0) {
3         return a;
4     }
5     return pgcd(b, a % b);
6 }
7
8 public static int pgcdIter(int a, int b) {
9     int aux;
10    while (b != 0) {
11        aux = a;
12        a = b;
13        b = aux % b;
14    }
15    return a;
16 }
```

**Exercice 2.** *Fibonacci : version récursive terminale.*

1. Donner un algorithme récursif terminal pour calculer la suite de Fibonacci.
2. Donnez une implémentation itérative de l'algorithme ci-dessus.

**Correction :**

```
1 public static long fibo(int n) {
2     if (n == 0 || n == 1) {
3         return 1;
4     }
5     return fibo(n - 1) + fibo(n - 2);
6 }
7
8 public static long fiboAux(int n, long acc1, long acc2) {
9     if (n == 0 || n == 1) {
10        return acc2;
11    }
12    return fiboAux(n-1, acc2, acc1 + acc2);
13 }
14
15 public static long fiboTerm(int n) {
16    return fiboAux(n, 1, 1);
17 }
```

**Exercice 3.** *Echarpes : version récursive terminale.*

On revient sur l'exercice des écharpe pour le Racing qui alternent bandes de couleurs blanc ou bleu qui peuvent être simples, doubles ou triples.

1. Écrivez une version récursive terminale `echarpesR92Terminale`.
2. Donnez une implémentation itérative de l'algorithme ci-dessus.

**Correction :**

```
1 static int echarpesR92TerminaleAux(int n) {
2     int a1, int a2, int a3) {
3     if (n == 1) {
4         return a1;
5     } else {
6         return echarpesR92TerminaleAux(n - 1, a2, a3, a1 + a2 + a3);
7     }
8 }
9
10 static int echarpesR92Terminale(int n) {
11     return echarpesR92TerminaleAux(n, 2, 4, 6);
12 }
13
14 static int echarpesR92TerminaleIterative(int n) {
15     int a1 = 2, a2 = 4, a3 = 6;
16     while (n != 1) {
17         n--;
18         int tmp = a1 + a2 + a3;
19         a1 = a2;
20         a2 = a3;
21         a3 = tmp;
22     }
23     return a1;
24 }
```