

# TD N°03 JAVA

## Exercise 1 :

```
public class Puissance {  
  
    /**  
     * Puissance d'un nombre dans la décomposition d'un autre.  
     * Par exemple, la puissance de 2 dans la décomposition de 36 est 2  
     * car  $18 = 2^2 \times 9$ .  
     */  
    public static void main(String[] args) {  
        System.out.print("Nombre à décomposer : ");  
        int n = Console.readInt();  
        int n0 = n;  
        System.out.print("Nombre de la décomposition : ");  
        int m = Console.readInt();  
        int puissance = 0;  
        while (n % m == 0) {  
            // Jamais exécuté si m ne divise pas n  
            puissance++;  
            n = n / m;  
        }  
        System.out.println("La puissance de " + m + " dans la décomposition de  
" + n0  
            + " est " + puissance);  
        if (puissance == 0) {  
            System.out.println("car " + n0 + " n'est pas divisible par " + m);  
        }  
        else {  
            System.out.println("car " + n0 + " = " + m + " ^ " + puissance + " x  
" + n);  
        }  
    }  
}
```

**Exercise 2 :**

```
public class Factorielle {  
  
    /**  
     * Calcul de n!.  
     * Version itérative.  
     */  
    public static void main(String[] args) {  
        System.out.print("Nombre n : ");  
        int n = Console.readInt();  
  
        // Version avec "tant que" (on pourrait aussi donner une variante  
        // qui collerait mieux à la boucle for avec incrémentation  
        // de i à la fin de la boucle).  
        // Possible aussi de donner une version qui décromente i  
        // (on part de n pour arriver à 1).  
        int i = 0;  
        int factorielle = 1;  
        while (i < n) {  
            // Invariant de boucle : factorielle = i!  
            i++;  
            factorielle *= i;  
            // Décommenter pour voir  
            // System.out.println(i + "! = " + factorielle);  
        }  
        System.out.println(n + "! = " + factorielle);  
  
        // Version avec "for"  
        // Si on veut prouver cette boucle il faut la transformer d'abord  
        // en boucle while pour que l'incrément de i2 soit explicite.  
        factorielle = 1;  
        for (int i2 = 1; i2 <= n ; i2++) {  
            factorielle *= i2;  
        }  
        System.out.println(n + "! = " + factorielle);  
    }  
}
```