
Exercice 1. Conversions implicites : *promotion numérique* et *conversion d'ajustement*

Soit ces déclarations :

```
byte b1 = 10, b2 = 20 ;
short p = 200 ;
int n = 500 ;
long q = 100 ;
float x = 2.5f ;
double y = 5.25 ;
```

Donner le type de conversion implicite ainsi que le type et la valeur des expressions arithmétiques suivantes :

```
b1+b2
p+b1
b1*b2
q+p*(b1+b2)
x+q*n
b1*q/x
b1*q*2./x
b1*q*2.f/x
```

Même exercice avec les déclarations et expressions suivantes :

```
char c = 60, ce = 'e', cg = 'g' ;
byte b = 10 ;

c + 1
2 * c
cg - ce
b * c
```

Exercice 2. La norme IEEE 754 pour les flottants

En Java, aucune opération sur les flottants ne conduit à l'arrêt de l'exécution du programme en cours. En revanche les nombres flottants respectent les conventions IEEE 754 qui imposent l'existence d'une représentation pour les valeurs « +l'infini », « -l'infini » et « non-calculable ».

Soit ces déclarations :

```
double x1 = 1e200, x2 = 1e210 ;
double y, z ;
```

Qu'affichent ces lignes ? :

```
y = x1 * x2 ;
x2 = x1 ;
z = y / (x2-x1) ;
System.out.println(y + " divise par " + (x2-x1) + " = " + z) ;
y = 1 ;
z = y / (x2-x1) ;
System.out.println(y + " divise par " + (x2-x1) + " = " + z) ;
```

```
z = (x2-x1) / (x2-x1) ;
System.out.println((x2-x1) + " divise par " + (x2-x1) + " = " + z) ;
System.out.println(z+ " + " + 1 + " = " + (z+1)) ;
```

Exercice 3. Les opérateurs logiques à « court-circuit » ou pas

En C, il n'existe que les opérateurs logiques ***et*** (&&) et ***ou*** (||) à court-circuit. Le second opérande du && (*resp. du ||*) n'est évalué que si le premier est vrai (*resp. faux*).

En Java, il existe les opérateurs logiques ***et*** (&&) et ***ou*** (||) à court-circuit et les opérateurs logiques ***et*** (&) et ***ou*** (|) sans court-circuit.

En déduire les résultats fournis par le programme suivant :

```
public class CouCir
{
   public static void main(String args[])
   {
      int i=10, j=5 ;
      if(i<5 && j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;

      if(i<5 & j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;

      if(i<15 && j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;

      if(i<15 || j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;

      if(i<5 || j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;

      if(i<15 | j++<10) System.out.println("vrai");
      else System.out.println("faux");

      System.out.println("i=" + i + " ; j=" + j) ;
   }
}
```

Exercice 4. Les structures de contrôle sans étiquette

Quels résultats fournit le programme suivant ? :

```
public class CouCir
{
   public static void main(String[] args)
   {
      int n=0 ;
      do
      {
         if(n%2==0)
         {
            System.out.println(n + " est pair") ;
            n += 3 ;
            continue ;
         }

         if(n%3==0)
         {
            System.out.println(n + " est multiple de 3") ;
            n += 5 ;
         }

         if(n%5==0)
         {
            System.out.println(n + " est multiple de 5") ;
            break ;
         }
         n += 1 ;
      }while(true) ;
   }
}
```

Exercice 5. *En cours (live coding).* Les entrées-sorties clavier et les structures de contrôle avec étiquette

Ecrire un programme qui demande à l'utilisateur de saisir un montant en euro entier (pas de centimes) et qui affiche toutes les manières possibles d'obtenir cette somme avec des pièces de 5 cents, 10 cents, 20 cents, 50 cents et 1 euro. Proposer à l'utilisateur de continuer ou de quitter après chaque solution trouvée : on pourra utiliser une instruction *break* à étiquette.