

TD1 (2h)**Exercice 1: Evaluation d'expressions arithmétiques**

On considère les déclarations suivantes :

```
byte b1 = 5, b2 = 30 ;
short s = 300 ;
int n = 600 ;
long q = 1500 ;
float x = 1.5E2f ;
double y = 5.0E3 ;
```

Donner le type et la valeur des expressions arithmétiques suivantes :

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

Exercice 2 : Conversion automatique du type char vers le type int

On considère les déclarations suivantes : `char c = 50, ci = 'i', ck = 'k' ;`
`byte b = 20 ;`

Donner le type et la valeur des expressions arithmétiques suivantes :

```
c + 1
2 * c
ck - ci
b * c
```

Exercice 3 : Quelles valeurs contiennent les variables n, p et q après évaluation de chaque expression?

```
int n = 3, p = 5, q = 7;
n = p = q = 10;
n += p += q;
q = n < p ? n++ : p++;
q = n > p ? n++ : p++;
n = q==p ? ++p : --q;
```

Exercice 4 : Sélecteur switch

On considère un entier `n` dont la valeur sera lue au clavier. Quel sera l'affichage à l'issue du switch pour chacune des valeurs saisies suivantes: 0, 1, 2, 3, 4, 10, -5?

```
switch (n) {
    case 0 : System.out.print ( "Bonjour, ");
    case 1 :
    case 2 : System.out.println ( "ça va?");
               break;
    case 3 : break;
    case 4 :
    case 5 : System.out.println ( "Hello!");
    default : System.out.println ( "Au revoir");
}
```

Exercice 5: Instructions itératives

a) Division Euclidienne

On se donne deux entiers positifs a et b , $b \neq 0$.

1- Ecrire le code permettant d'effectuer la division euclidienne de a par b en utilisant l'algorithme suivant :

```
int q = 0, r = a;
Tant que r >= b
    remplacer q par q + 1
    remplacer r par r - b
```

En sortie, q est le quotient et r le reste de la division euclidienne de a par b .

On souhaite afficher les calculs intermédiaires comme dans l'exemple :

```
11 = 4 * 0 + 11
11 = 4 * 1 + 7
11 = 4 * 2 + 3
```

2- Ecrire le code permettant d'effectuer la division euclidienne de a par b en utilisant l'algorithme suivant :

```
int q = 0, r = a;
Tant que r >= b
    trouver le plus grand entier n tel que  $10^n b \leq r$ 
    trouver le plus grand chiffre c tel que  $10^n bc \leq r$ 
    remplacer q par q +  $10^n c$ 
    remplacer r par r -  $10^n bc$ 
```

On souhaite également afficher les calculs intermédiaires.

Note : Pour calculer 10^n , appeler la fonction `Math.pow(10, n)` qui retourne un double.

b) Calcul de la valeur d'une série

On se donne un entier positif n .

Ecrire le code permettant de calculer la somme des n premiers termes de la série harmonique :

$1 + 1/2 + 1/3 + \dots + 1/n$.

Le résultat sera stocké dans une variable de type `float`.

Exercice 6:

```
public class Test1Tableau {

    public static void main(String[] args) {
        int[] t1 = {3, 7, 24, 30} ;
        int[] t2= new int[10];
        for ( int i = 0; i<10; i++) {
            t2[i]=5*i;
        }
        t2 = t1;
        for ( int i = 0; i<10; i++) {
            System.out.println(t2[i]);
        }
    }

}
```

Qu'obtient-on lorsqu'on exécute ce programme ?

Exercice 7:

```

public class Test2Tableau {

    public static void main(String[] args) {
        int[][] t = new int[3][];
        for ( int i = 0; i<3; i++) {
            t[i] = new int[i+1];
            for ( int j = 0; j<t[i].length; j++)
                t[i][j] = i+j;
        }
        for ( int i = 0; i<3; i++) {
            System.out.print("Tableau numéro " + i + " = ");
            for ( int j = 0; j<t[i].length; j++)
                System.out.print( t[i][j] + " ");
            System.out.println( );
        }
    }
}

```

Qu'obtient-on lorsqu'on exécute ce programme ?

Exercice 8: On considère un tableau rempli avec les notes d'une classe :

```
double[] lesNotes.
```

- Ecrire le code permettant de calculer la moyenne de ces notes.
- Ecrire le code permettant de comptabiliser le nombre de notes supérieures ou égales à la moyenne.

Exercice 9 : On considère un tableau d'entiers : `int[] tab.`

Ecrire le code permettant de tester si ce tableau est un palindrome, c'est à dire qu'il se lit de la même manière dans les deux sens.

Exercice 10 : Tri par sélection

On considère un tableau d'entiers : `int[] tab.`

- Ecrire le code permettant de rechercher le premier indice `min` du plus petit élément du tableau.
- Compléter le code précédent sachant que pour effectuer un tri par sélection, on recherche l'indice `min` du plus petit élément du tableau puis on échange `tab[0]` avec `tab[min]`. On recommence ces deux opérations avec les éléments du tableau à partir de la l'indice 1. Et ainsi de suite.

Exercice 11 : Tri par insertion

On considère un tableau d'entiers : `int[] tab.`

- Ecrire le code permettant d'effectuer un tri par insertion. Le tri par insertion procède de la même façon qu'un joueur de cartes pour trier ses cartes. On prend une carte, puis 2 et on les met dans l'ordre si nécessaire, puis 3 et on met la 3ème carte à sa place dans les 2 premières,...De manière générale, on suppose les $i-1$ premières cartes triées. On prend la i ème carte, et on essaie de la mettre à sa place dans les $i-1$ cartes déjà triées.
- Pour classer le i ème élément du tableau `tab`, on regarde successivement en marche arrière à partir du $i-1$ ième. On décale les éléments visités vers la droite pour pouvoir mettre `tab[i]` à sa juste place.