



Corrigé du TP de Java n°4

1 ECRIRE DES FONCTIONS EN JAVA

1.

```
import java.util.Random;

public class Main
{
    public static void AfficherTableau(int tab[])
    {
        int compteur;
        for(compteur=0; compteur<tab.length; compteur=compteur+1)
        {
            System.out.println(tab[compteur]);
        }
    }

    public static void main(String args[])
    {
        Random m = new Random();

        //Construction du premier tableau
        int hasard = m.nextInt(20)+1;
        int tabl[];
        tabl = new int[hasard];
        int x=0;
        while(x<tabl.length)
        {
            hasard = m.nextInt(101);
            tabl[x] = hasard;
            x=x+1;
        }

        //Et on affiche le tableau grâce à notre fonction
        AfficherTableau(tabl);
    }
}
```

2.

```
import java.util.Random;

public class Main
{
    public static void AfficherTableau(int tab[])
    {
```

```
int compteur;
for(compteur=0; compteur<tab.length; compteur=compteur+1)
{
    System.out.println(tab[compteur]);
}

public static int PositionPlusGrandElement(int tab[])
{
    int compteur;
    int position_max = 0;
    for(compteur=1; compteur<tab.length; compteur=compteur+1)
    {
        if(tab[position_max] < tab[compteur])
        {
            position_max = compteur;
        }
    }
    return position_max;
}

public static void main(String args[])
{
    Random m = new Random();

    //Construction du premier tableau
    int hasard = m.nextInt(20)+1;
    int tabl[];
    tabl = new int[hasard];
    int x=0;
    while(x<tabl.length)
    {
        hasard = m.nextInt(101);
        tabl[x] = hasard;
        x=x+1;
    }

    //Et on affiche le tableau grâce à notre fonction
    AfficherTableau(tabl);

    //Et on affiche la position du plus grand élément
    int p = PositionPlusGrandElement(tabl);
    System.out.println("Position du plus grand élément :");
    System.out.println(p);
}
}
```

3. Si on augmente le nombre de cases d'un facteur n , le temps de calcul doit augmenter d'un facteur n . Par exemple, si la fonction s'exécute en 2s sur un tableau de 100 000 cases, il mettra 20s sur un tableau de 1 000 000 cases (10 fois plus longtemps sur un tableau 10 fois plus grand).

4. On modifie la fonction `PositionPlusGrandElement` afin qu'elle prenne deux paramètres : le tableau, et un entier n qui sera le nombre de cases que l'on souhaite que la fonction prenne en compte lorsqu'elle recherchera le plus grand élément : la fonction n'ira pas regarder dans le tableau au delà des n premières cases.

Le tri consistera à chercher le plus grand élément sur un tableau qui devient toujours plus petit, et de placer ce plus petit élément en fin de tableau.

```
import java.util.Random;

public class Main
{
    public static void AfficherTableau(int tab[])
    {
        int compteur;
        for(compteur=0; compteur<tab.length; compteur=compteur+1)
        {
            System.out.println(tab[compteur]);
        }
    }

    public static int PositionPlusGrandElement(int tab[], int n)
    {
        int compteur=0;
        int position_max = 0;
        while(compteur < tab.length && compteur < n)
        {
            if(tab[position_max] < tab[compteur])
            {
                position_max = compteur;
            }
            compteur=compteur+1;
        }
        return position_max;
    }

    public static void TrierTableau(int tab[])
    {
        int compteur, position_max, c;
        for(compteur=tab.length-1; compteur>=0; compteur=compteur-1)
        {
            position_max = PositionPlusGrandElement(tab, compteur+1);
            c=tab[position_max];
            tab[position_max]=tab[compteur];
            tab[compteur]=c;
        }
    }

    public static void main(String args[])
    {
        Random m = new Random();

        //Construction du premier tableau
        int hasard = m.nextInt(20)+1;
        int tabl[];
        tabl = new int[hasard];
        int x=0;
        while(x<tabl.length)
        {
            hasard = m.nextInt(101);
            tabl[x] = hasard;
            x=x+1;
        }

        //Et on trie le tableau
        TrierTableau(tabl);

        //Et on affiche le tableau grâce à notre fonction
        AfficherTableau(tabl);
    }
}
```

```
}  
}
```

5. Si on augmente le nombre de cases d'un facteur n , le temps de calcul doit augmenter d'un facteur n^2 . Par exemple, si la fonction s'exécute en 2s sur un tableau de 100 000 cases, il mettra 200s sur un tableau de 1 000 000 cases (100 fois plus longtemps sur un tableau 10 fois plus grand).

2 TABLEAUX À DEUX DIMENSIONS

1.

```
import java.util.Random;  
public class Main  
{  
    public static void main(String args[])  
    {  
        Random m = new Random();  
        int hasard = m.nextInt(8);  
    }  
}
```

2.

```
import java.util.Random;  
public class Main  
{  
    public static void main(String args[])  
    {  
        Random m = new Random();  
        int hasard = m.nextInt(8)+1;  
    }  
}
```

3.

```
public class Main  
{  
    public static void main(String args[])  
    {  
        int tab[][];  
        tab = new int[7][7];  
    }  
}
```

4.

```
import java.util.Random;  
public class Main  
{  
    public static void main(String args[])  
    {  
        Random m = new Random();  
        int la = m.nextInt(8)+1;  
        int lb = m.nextInt(8)+1;  
        int tab[][];
```

```
        tab = new int[la][lb];
    }
}
```

5.

```
import java.util.Random;
public class Main
{
    public static void main(String args[])
    {
        Random m = new Random();
        int la = m.nextInt(8)+1;
        int lb = m.nextInt(8)+1;
        int tab[][];
        tab = new int[la][lb];

        int cpta, cptb;
        for(cpta=0; cpta<la; cpta=cpta+1)
        {
            for(cptb=0; cptb<lb; cptb=cptb+1)
            {
                tab[cpta][cptb]= m.nextInt(101)+100;
            }
        }
    }
}
```

6.

```
import java.util.Random;
public class Main
{
    public static void AfficherTableau(int tab[][])
    {
        int i,j;
        for(i=0; i<tab.length; i=i+1)
        {
            for(j=0; j<tab[i].length; j=j+1)
            {
                System.out.print(tab[i][j]);
                System.out.print(" ");
            }
            System.out.println();
        }
    }

    public static void main(String args[])
    {
        Random m = new Random();
        int la = m.nextInt(8)+1;
        int lb = m.nextInt(8)+1;
        int tab[][];
        tab = new int[la][lb];

        int cpta, cptb;
        for(cpta=0; cpta<la; cpta=cpta+1)
        {
            for(cptb=0; cptb<lb; cptb=cptb+1)
            {
                tab[cpta][cptb]= m.nextInt(101)+100;
            }
        }
    }
}
```

```
        }
    }

    AfficherTableau(tab);
}
}
```

7.

```
import java.util.Random;
public class Main
{
    public static void AfficherPositionPlusGrandElement(int tab[][])
    {
        int maxa, maxb, i, j;
        maxa=0;
        maxb=0;
        for(i=0; i<tab.length; i=i+1)
        {
            for(j=0; j<tab[i].length; j=j+1)
            {
                if(tab[i][j] > tab[maxa][maxb])
                {
                    maxa=i;
                    maxb=j;
                }
            }
        }
        System.out.println("Le plus grand élément est en");
        System.out.println(maxa);
        System.out.println(maxb);
    }

    public static void AfficherTableau(int tab[][])
    {
        int i, j;
        for(i=0; i<tab.length; i=i+1)
        {
            for(j=0; j<tab[i].length; j=j+1)
            {
                System.out.print(tab[i][j]);
                System.out.print(" ");
            }
            System.out.println();
        }
    }

    public static void main(String args[])
    {
        Random m = new Random();
        int la = m.nextInt(8)+1;
        int lb = m.nextInt(8)+1;
        int tab[][];
        tab = new int[la][lb];

        int cpta, cptb;
        for(cpta=0; cpta<la; cpta=cpta+1)
        {
            for(cptb=0; cptb<lb; cptb=cptb+1)
            {
                tab[cpta][cptb]= m.nextInt(101)+100;
            }
        }
    }
}
```

```
        }  
    }  
  
    AfficherTableau(tab);  
    AfficherPositionPlusGrandElement(tab);  
}  
}
```

Si l'on voulait renvoyer la position du plus grand élément, on ferait face à un problème : on ne peut renvoyer qu'une seule valeur entière en sortie d'une fonction. Or, la position du plus grand élément tient sur deux entiers.