

*IF1: Introduction à l'Informatique et à la programmation - Devoir sur Table*  
(Correction)

**Documents non autorisés. Le barème est donné seulement à titre indicatif.**

**Exercice 1 (10 points)**

1. *Écrire une fonction Java `hasard` qui génère au hasard un entier entre 1 et 1000.*

*Correction :*

```
static int hasard(){
    return partieEntiere(1000*Math.random()+1);
}
```

2. *Écrire une fonction Java `miroir` qui prend un entier positif  $n$  en argument et qui renvoie son miroir. Ainsi par exemple, l'appel de fonction `miroir(98365964)` renvoie l'entier 46956389.*

*Correction :*

```
static int miroir (int n){
    int res;
    int a = n;
    if (n>=0){
        res=0;
        while (a !=0){
            res = res*10 + (a %10);
            a = a/10; }
        return res; }
    else return -1;
}
```

3. *Écrire un programme Java `DevinerMiroir` qui génère au hasard un entier  $n$  entre 1 et 1000, puis fait deviner son miroir à l'utilisateur en précisant, après chaque essai infructueux, si  $n$  est supérieure ou inférieure au nombre à trouver. Voici un exemple d'exécution:*

```
> java DevinerMiroir
Rentrez un nombre entre 1 et 1000 : 500
Le nombre a trouver est plus grand.
Rentrez un nombre entre 1 et 1000 : 950
Le nombre a trouver est plus grand.
Rentrez un nombre entre 1 et 1000 : 975
Le nombre a trouver est plus petit.
Rentrez un nombre entre 1 et 1000 : 965
Le nombre a trouver est plus grand.
Rentrez un nombre entre 1 et 1000 : 970
Le nombre a trouver est plus petit.
```

Rentrez un nombre entre 1 et 1000 : 966  
Le nombre a trouver est plus grand.  
Rentrez un nombre entre 1 et 1000 : 967  
Vous avez trouvé le miroir de 769.

4. *Raffiner le programme précédent pour qu'il affiche au final le nombre total d'essais nécessaires à l'utilisateur pour gagner.*

**Correction :**

```
import java.util.Scanner;

class DevinerMiroir{

    static int partieEntiere(double x){
        int e = 0;
        while(e <= x) e++;
        return e - 1;}

    static int hasard(){
        return partieEntiere(1000*Math.random()+1);
    }

    static int miroir (int n){
        int res;
        int a = n;
        if (n>=0){
            res=0;
            while (a !=0){
                res = res*10 + (a %10);
                a = a/10; }
            return res; }
        else return -1;
    }

    public static void main(String[] args){
        int r, essais;
        int a,b;
        boolean reponse;

        a = hasard();
        b= miroir(a);
        reponse = false;
        essais = 0;
        while (!reponse){
            System.out.print("Rentrez un nombre entre 1 et 1000 : ");
            Scanner sc = new Scanner(System.in);
            r = sc.nextInt();
            essais = essais + 1;
            if (r == b){
                System.out.println("Vous avez trouve le miroir de "+a+" en "+essais+ " coups");
                reponse = true;}
            else if (r > b)
                System.out.println("Le nombre a trouver est plus petit.");
```

```

        else System.out.println("Le nombre a trouver est plus grand.");}
    }
}

```

**Exercice 2 (10 points)** *On veut calculer un polynôme de la forme  $a_0 \cdot x^0 + a_1 \cdot x^1 + \dots + a_n \cdot x^n$ .*

1. *Écrire une fonction polynome qui prend en argument une valeur réelle  $x$  et un tableau d'entiers  $a$  contenant les valeurs de  $a_0, \dots, a_n$  et qui renvoie la valeur du polynôme.*

**Correction :**

```

    public static double polynome(int[] a, double x){
double res=0;
    for (int i=0; i<a.length;i++)
res=res+(a[i] *Math.pow(x,i));
    return res; }

```

2. *Écrire une fonction modifier de type void qui prend en argument un tableau d'entiers  $a$  et qui modifie ce même tableau en multipliant chacun de ses éléments par 5.*

**Correction :**

```

    public static void modifier(int[] a){
    for (int i=0; i<a.length;i++)
a[i]= a[i]*5; }

```

3. *Écrire le programme principal qui demande d'abord à l'utilisateur de rentrer un entier  $n$ , puis de rentrer les  $n$  valeurs d'un tableau  $t$ , et enfin la valeur de  $y$ .*

*Le programme imprime ensuite le résultat de  $\text{polynome}(t,y)$ , puis modifie le tableau  $t$  en utilisant la fonction `modifier`, puis imprime à nouveau  $\text{polynome}(t,y)$ .*

**Correction :**

```

import java.util.*;

public class Polynome{

    public static double polynome(int[] a, double x){
double res=0;
    for (int i=0; i<a.length;i++)
res=res+(a[i] *Math.pow(x,i));
    return res;}

    public static void modifier(int[] a){
    for (int i=0; i<a.length;i++)
a[i]= a[i]*5; }

    public static void main(String[] args){
Scanner sc = new Scanner(System.in);
System.out.print("Taille tableau : ");
int n=sc.nextInt();

```

```
int[] t= new int[n];
for (int i=0;i<n;i++){
    System.out.print("Un element : ");
    t[i]=sc.nextInt();}
    System.out.print("Valeur de y : ");
int y=sc.nextInt();

    System.out.println("Resultat : "+polynome(t,y));
modifier(t);
    System.out.println("Resultat : "+polynome(t,y)); }
}
```

4. *Donner un exemple d'exécution du programme principal.*