

# TP 6: Boucles et tableaux. Corrigé

Informatique Fondamentale (IF121)

16 Novembre 2004

## 1 Suites numériques : encore et toujours

### Exercice 1 (Somme des puissances $k^{ime}$ )

On rappelle la méthode puissance (sur les entiers cette fois) et on écrit la somme :

```
static int puissance (int nombre, int k){
    int p=1;
    if (k >= 0){
        for (int i=1; i <= k; i++){
            p = p*nombre;
        }
    }
    else{
        for (int i=1; i <= ((-1)*k); i++){
            p = p/nombre;
        }
    }
    return p;
}

static double sommePuissances (int k, int n){
    int res=0;
    for (int i=1; i <=n; i++) res = res+puissance(i,k);
    return res;
}
```

### Exercice 2 (Une suite)

```
import fr.jussieu.script.Deug;
class Suite{
    static double Un(double a, double b, int n){
        double x=b;
        for (int i=1; i <= n; i++) x = ((x*x+a)/(2*x));
        return x;
    }

    public static void main (String[] args){
        Deug.println("veuillez entrer a");
        double a = Deug.readDouble();
        Deug.println("veuillez entrer b");
        double b = Deug.readDouble();
        Deug.println("veuillez entrer n");
        int n = Deug.readInt();
        Deug.println("une approximation de la racine de "+a+" est "+Un(a,b,n));
    }
}
```

### Exercice 3 (Deux Suites)

```
import fr.jussieu.script.Deug;
class SuiteDouble{
    static double approximationDePi (int n){
        double u = 1.0;
        double v = 2.0;
```

```

        for (int i=1; i <= n; i++){
            u = (u+v) /2;
            v = Math.sqrt(u*v);
        }
        double pi = Math.sqrt(27.0) /v;
        return pi;
    }

    public static void main (String[] args){
        Deug.println("veuillez entrer un entier n d'après lequel calculer une valeur approchée de Pi");
        int n = Deug.readInt();
        Deug.println("une approximation de Pi à partir de v"+n+" est: "+approximationDePi(n));
    }
}

```

## 2 Traitement de données

### Exercice 4 (Recherche de "en")

```

import fr.jussieu.script.Deug;
class Rechercheen{
    static int compte(String s){
        int res=0;
        for (int i=0; i < Deug.length(s)-1; i++){
            if ((Deug.charAt(s,i)=='e')&&(Deug.charAt(s,i+1)=='n')) res++;
        }
        return res;
    }
    public static void main (String[] args){
        Deug.println("veuillez entrer le texte");
        String s = Deug.readLine();
        Deug.println("Il y a "+compte(s)+" en dans le texte.");
    }
}

```

### Exercice 5 (Palindromes)

```

import fr.jussieu.script.Deug;
class Palindrome{
    public static boolean estPalindrome (String mot) {
        int i = 0;
        boolean b=true;
        while ((i<Deug.length(mot)/2)&& b) {
            b=(Deug.charAt(mot, i) == Deug.charAt(mot, n - 1 - i));
            i++;
        }
        return b;
    }
    public static void main (String[] args) {
        Deug.print("Entrez un mot : ");
        String mot = Deug.readLine();
        if (estPalindrome(mot)) Deug.println(mot + " est un palindrome.");
        else Deug.println(mot + " n'est pas un palindrome.");
    }
}

```

## 3 Tableaux de nombres

### Exercice 6 (Norme, produit scalaire)

Pour la lecture du tableau il faut penser à initialiser le tableau, puis il s'agit d'une boucle for. Pour le produit scalaire on parcourt les deux tableaux en même temps. La norme est le produit scalaire du tableau par lui même. Cela donne le code suivant :

```

import fr.jussieu.script.Deug;
class Tableau{
    static double[] tableau (int n){
        double[] t = new double[n];
    }
}

```

```

    for (int i=0; i < n; i++){
        Deug.print("R el n" + (i+1) + " = ");
        t[i] = Deug.readDouble();
    }
    return t;
}

static double produitScalaire (double[] x, double[] y){
    int n = x.length;
    double prod = 0.0;
    for (int i = 0; i < n; i++){
        prod = prod + x[i]*y[i];
    }
    return prod;
}

static double norme (double[] x){
    return Maths.sqrt(produitScalaire(x,x));
}

public static void main (String[] args){
    Deug.print("Dimension de l'espace vectoriel ?");
    int n = Deug.readInt();
    Deug.print("Vecteur t1 ?");
    double[] t1 = tableau(n);
    Deug.print("Vecteur t2 ?");
    double[] t2 = tableau(n);
    Deug.println("Produit scalaire = " + produitScalaire(t1,t2));
    Deug.println("Norme de t1 = " + norme(t1));
    Deug.println("Norme de t2 = " + norme(t2));
}
}

```

### Exercice 7 (Polyn mes)

On rappelle   nouveau la fonction puissance, sur un double comme dans le TP5.

```

public static double puissance (double x, int n) {
    double y = 1;
    for (int i = 0; i < n; i++) y = y * x;
    return y;
}

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

public static double horner (double[] a, double x) {
    double res = 0;
    for (int i = a.length - 1; i >= 0; i--) res = res * x + a[i];
    return res;
}
}

```

### Exercice 8 (Constitution d'un histogramme)

L'id e est de cr er un tableau qui va contenir la r partition par note (la case  $i$  indiquera le nombre d' l ves ayant la note  $i$ ) et de parcourir le tableau des notes en mettant   jour celui de la r partition. On a alors :

```

static int[] tableauStats (int[] notes){
    int[] stats = new int[11];
    int note;
    for (int i=0; i < notes.length ; i++){
        note = notes[i];
        stats[note] ++;
    }
    return stats;
}
}

```