

Examen de TP - Groupe A4

Informatique Fondamentale (IF121)

15 Décembre 2004

1 Algèbre

Exercice 1 : *Division entière*

```
- static int diviser(int k,int n){
    int res=0;
    while (res*n<=k) res++;
    return res-1;
}
- static int modulo(int k,int n){
    int res=0;
    while (res*n<=k) res++;
    return k-(res-1)*n;
}
```

Exercice 2 : *Nombres premiers*

```
- static boolean isprime(int n){
    int k=2;
    boolean b=true;
    while (b&&(k< n)) {b=(modulo(k,n)!=0);k++;}
    return b;
}
- static void prime(int n){
    for (int i=1;i<=n;i++){
        if (isprime(i)) Deug.println(i);
    }
}
```

Exercice 3 : *Suite pour calculer π*

```
import fr.jussieu.script.Deug;

class CalculDePi{

    static double Un(double a,int n){
        if (n<=0) return (a*a);
        else return (a*a)/(6+Un(a+2,n-1));
    }

    public static void main(String[] args){
        Deug.println(Un(1,Deug.stringToInt(args[0]))+3);
    }
}
```

2 Graphiques

Exercice 4 : Boîtes imbriquées

```
import fr.jussieu.script.Deug;
class Dessin{

    static void afficheBoite (int x, int y, int taille) {
        Deug.drawLine (x, y, x, y+taille);
        Deug.drawLine (x, y, x+taille, y);
        Deug.drawLine (x, y+taille, x+taille, y+taille);
        Deug.drawLine (x+taille, y, x+taille, y+taille);
    }

    public static void main (String[] args) {
        int x = Deug.stringToInt (args[0]),
            y = Deug.stringToInt (args[1]),
            n = Deug.stringToInt (args[2]),
            t = Deug.stringToInt (args[3]);
        Deug.startDrawings(800, 600);
        for (int i = 0 ; i < n ; i++) {
            afficheBoite (x, y, t);
            x -= 10;
            y -= 10;
            t += 20;
        }
    }
}
```

3 Tableaux

Exercice 5 : Recherche

```
static boolean rechercheTableau(char[] tab, char c, int a, int b) {
    if (a >= tab.length) return false;
    int i = a;
    while ((i <= b) && (i < tab.length) && (tab[i] != c)) i++;
    return !((i == tab.length) || (i > b));
}

static int nombreOccurrences (char[] tab, char c) {
    int compteur = 0;
    for (int i = 0 ; i < tab.length ; i++) {
        if (rechercheTableau (tab, c, i, i)) compteur++;
    }
    return compteur;
}
```

Exercice 6 : Décalage

```
static void decal1 (int[] tab) {
    int caseSuivante = tab[0], tmp;
    for (int i = 1; i < tab.length; i++) {
        tmp = tab[i];
        tab[i] = caseSuivante;
        caseSuivante = tmp;
    }
    tab[0] = caseSuivante;
}
```

```

static void decalN (int[] tab, int n) {
    if (n > 0) {
        for (int i = 0 ; i < n; i++) decal1(tab);
    }
}

```

Exercice 7 : *Moyenne olympique*

```

static double moyenneOlympique (int[] tab) {
    int min=0, max=0, somme=tab[0];
    for (int i = 1 ; i < tab.length ; i++) {
        somme += tab[i];
        if (tab[i] > tab[max]) max = i;
        if (tab[i] < tab[min]) min = i;
    }
    return (double)(somme-tab[min]-tab[max])/((double)(tab.length-2));
}

```

Exercice 8 : *Recherche dichotomique*

```

static int rechercheDicho(double[] tab,double n){
    int a=0;
    int b=tab.length;
    int k;
    while ((b-a)>1){
        k=(a+b)/2;
        if (n<=tab[k]) b=k;
        else a=k;
    }
    return a;
}

```