

Université Paris 7 - Denis Diderot
CI2 — Paritel 3 mars 2007 — Durée : 2 heures
Documents, caleulottes, ordinateurs et téléphones non autorisés.

Exercice 1 — Appels par valeurs, appels par références

Quel sont tous les résultats affichés par le code suivant ?

```
class IntRef {
    public int valeur;
    public IntRef(int n) {valeur = n;}
}
class appelsValRef{
    static int f1(int x){
        System.out.println("dans f1 : x="+x);
        x= x+1;
        System.out.println("dans f1 : x="+x);
        return x;
    }
    static int f2(IntRef xRef){
        xRef.valeur= xRef.valeur+1;
        return xRef.valeur;
    }
    static void f3(int[] T,int x, int y){
        T[1]=x;
        T[0]=y;
        System.out.println("dans f3 : T[0]="+T[0]);
    }
    public static void main(String[] param){
        int x=5;
        System.out.println("dans main : x="+x);
        f1(x);
        System.out.println("dans main : x="+x);
        IntRef xRef= new IntRef (x);
        System.out.println("dans main : xRef.valeur="+xRef.valeur);
        f1(xRef.valeur);
        System.out.println("dans main : xRef.valeur="+xRef.valeur);
        f2(xRef);
        System.out.println("dans main : xRef.valeur="+xRef.valeur);
        xRef.valeur= f2(xRef);
        System.out.println("dans main : xRef.valeur="+xRef.valeur);
        x= f2(xRef);
        System.out.println("dans main : xRef.valeur="+xRef.valeur);
        System.out.println("dans main : x="+x);
        int[] T= new int[2];
        T[0]= 2;
        T[1]= 4;
        f1(T[0]);
        System.out.println("dans main : T[0]="+T[0]);
        f3(T,T[0],T[1]);
        System.out.println("dans main : T[0]="+T[0]);
    }
}
```

Exercice 2 — Fonction récursive On considère la fonction des entiers vers les entiers définie par :

$$f(x) = \begin{cases} 1 & \text{si } x < 0 \\ 3 * f(x - 1) - f(x - 2) & \text{sinon} \end{cases}$$

1. Écrire en java une fonction récursive (naïve) prenant un entier x en argument, et renvoyant $f(x)$.
2. Dessiner l'arbre des appels récursifs pour le calcul de $f(3)$.
3. Donner la suite des appels dans l'ordre où ils sont exécutés pour le calcul de $f(3)$.
4. Écrire en java une version itérative de la fonction f qui utilise un tableau.
5. Dessiner le tableau qui est construit par le programme itératif pour le calcul de $f(3)$.
6. Écrire en java une version "programmation dynamique" récursive de la fonction f .
7. Dans quel ordre est rempli le tableau dans ce dernier cas? Quel est l'arbre des appels récursifs?
8. Parmi les trois versions proposées, quelle est la plus lente à l'exécution? Quelle est la plus rapide?

Exercice 3 — Empilement de boîtes On va définir une nouvelle méthode de calcul à partir de piles de boîtes. Chaque boîte possède une couleur (bleue ou rouge ici) et peut contenir des jetons (ou des grains de riz). Pour commencer le calcul, une seule boîte *bleue* est posée sur le sol, elle contient, par exemple, 7 jetons. On applique ensuite les règles suivantes, étape par étape.

- Si la boîte en haut de la pile est bleue, on compte le nombre de jetons qu'elle contient, et ensuite :
 - si elle contient 0 ou 1 jeton, on la retire de la pile et on vide son contenu dans la boîte suivante,
 - si elle contient au moins 2 jetons, on la remplace par une boîte rouge avec autant de jetons, et on empile dessus un nouvelle boîte bleue qui contient 2 jetons de moins.
- Si la boîte en haut de la pile est rouge, on la retire de la pile et on vide son contenu dans la boîte suivante (ou sur le sol si c'est la dernière).

1. Dessinez les étapes de successives de calcul quand on part d'une boîte bleue avec 7 jetons. La première étape peut être représentée comme :



(Bien sûr, on peut utiliser de vrais couleurs) Combien de jetons obtient-on à la fin?

2. Soit f la fonction telle que $f(n)$ est le nombre de jetons que l'on obtient à la fin du calcul si on part de n jetons dans une boîte bleue. Exprimer $f(n)$ en fonction de n , de $f(n - 1)$ et de $f(n - 2)$.
3. Écrire en java une méthode récursive qui prend en argument le nombre de jetons dans la première boîte bleue et qui retourne le nombre de jetons obtenus à la fin. La pile des appels récursifs de cette méthode devra ressembler à la pile des boîtes.
4. Écrire une méthode itérative qui donne le même résultat final. Justifiez.