

Examen de CI2 (Concepts Informatiques)
Licence Informatique - Première année - Seconde session

Vendredi 21 Juin 2013
Durée : 3 heures

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé - La copie ou les brouillons du voisin ne sont pas des supports autorisés.

Positionnez impérativement vos mobiles en mode « avion ». Le sujet fait 3 pages

1 Variables...

Exo1.java

```
1 public class Exo1 {  
2     public static int f(int n) {  
3         int []tab = new int[n+1];  
4         tab[0] = tab[1] = 1;  
5         for (int i=2; i<=n; i++) tab[i] = tab[i-1]*tab[i-2]+1;  
6         return tab[n];  
7     }  
8     public static void main(String []args) {  
9         System.out.println(f(4));  
10        System.out.println(f(5));  
11    }  
12 }
```

1. Quels sont les affichages produits à l'exécution du programme ?
2. Si on appelle la fonction `f` avec le paramètre $n > 1$,
 - (a) Quelle quantité de mémoire (au minimum) est utilisée ? Le résultat sera exprimé en nombre d'`int` nécessaires au calcul. Justifiez.
 - (b) Dessinez l'état de la mémoire juste avant l'exécution de l'instruction `return` pour $n = 5$. On distinguera pile et tas.

2 Récursion...

Exo2.java

```
1 public class Exo2 {  
2     public static int f(int n) {  
3         if (n==0 || n==1) return 1;  
4         return f(n-2)*f(n-1)+1;  
5     }  
6     public static void main(String []args) {  
7         System.out.println(f(5));  
8         System.out.println(f(6));  
9     }  
10 }
```

1. La fonction `f` est-elle récursive terminale. Justifiez.
2. Quels sont les affichages produits à l'exécution du programme ?
3. Dessinez l'arbre d'exécution (l'arbre des appels) pour le calcul de `f(5)`.
4. (Pas facile) Sauriez-vous déterminer le nombre d'appels total à la fonction `f()` pour tout n ?
5. Utilisez la technique de la fonction à mémoire pour réduire le nombre d'appels de la fonction.
6. Mêmes questions si l'on remplace `return f(n-2)*f(n-1)+1` par `return 1+f(n-1)*f(n-2)`. Justifiez.

3 Traduction...

Dans cet exercice, il s'agit de traduire le programme précédent (`Exo2.java`) en un autre programme Java de forme particulière étudiée en cours : le programme principal ne devra pas contenir d'autres variables qu'un entier de nom `instructionCourante` (ou un symbole plus « court » `ic` conviendra!), un tableau d'entiers de nom `memoire` et accessoirement une pile (dont il faudra détailler les caractéristiques si vous en utilisez une, on rappelle qu'en cours nous avons codé la pile des appels directement dans la mémoire). Le code sera simplement constitué d'une boucle `while` contenant un unique `switch` avec autant de cas que nécessaires. Dans le programme traduit, il est formellement interdit d'employer toute autre construction de haut-niveau : pas de boucle `for`, ni d'autre boucle `while`, ni d'autre `switch`, etc.

On impose qu'un appel de fonction utilise la pile de la façon suivante (c'est-à-dire comme il a été fait en cours) :

- l'appel s'occupe d'empiler ce qui sera nécessaire au bon fonctionnement du code appelé;
- sont empilés dans l'ordre : la valeur de retour (s'il y en aura une), l'adresse de retour, les paramètres dans l'ordre de leur déclaration;
- la fonction appelée s'occupe de dépiler tout ce dont on n'aura pas besoin au retour, c'est-à-dire, tout **sauf** la valeur de retour;
- c'est le code appelant qui s'occupe de dépiler la mémoire associée à la valeur de retour.

1. Quelle sera la taille minimale de la **mémoire** qui devra être employée dans le programme traduit ? L'unité de mesure est un `int`.
2. Précisez où les variables du programme d'origine seront stockées dans le programme traduit ?
3. Traduisez le programme comme étudié en cours.

4 Rebroussement...

On cherche à construire une fonction permettant, étant donné une carte routière (un graphe) connectant des villes (des nœuds), de déterminer s'il existe un chemin d'une ville à une autre de longueur plus petite qu'une quantité donnée (par exemple, existe-t'il un chemin de Paris à Marseille de moins de 900km?). Pour notre fonction le graphe est codé simplement par un tableau à deux dimensions, la valeur de la case i, j fournit la longueur du chemin de i à j , si cette valeur est nulle, par convention cela signifie que i et j ne sont pas connectés.

1. En quoi la technique du rebroussement est-elle adéquate pour ce problème?
2. Construisez un petit graphe (moins de 5 nœuds) dans lequel il vous sera possible d'illustrer le rebroussement. Indiquez pour quelles villes et quelle longueur un rebroussement est observable.
3. Représentez sous la forme d'un tableau à deux dimensions le graphe que vous avez construit à la question précédente.