Université Libanaise
Faculté des Sciences
Section I

I2204 Fr - INFO216 Fr
Programmation Impérative
Examen Partiel

Licence - Informatique
Durée : 60 minutes
21 Novembre 2017

Question 1 : Énigme mystérieuse (*6 pts*)
Dessiner l'état de la mémoire du programme ci dessous. En déduire la valeur de sortie.

```
#include <stdio.h>
int enigma(int n1, int n2) {
    if (n1 == 0) return 0;
    return n2 + enigma(n1 - 1, n2);
}
int mystery(int n) {
    if (n == 0) return 1;
    return enigma(2, mystery(n - 1));
}
void mysteryTest(){
    printf("%d",mystery(2));
}
void main(){
    mysteryTest();
}
```

Question 2 : Élever un nombre à une puissance de façon récursive (*6 pts*)
Il est facile d'implémenter une fonction itérative `raiseIntToPower` qui calcule n élevé à la$k$ ème puissance, n et k étant des entiers positifs.

```
#include <stdio.h>
void raiseIntToPowerTest();
int raiseIntToPower(int n, int k) {
    int result = 1;
    for (int i = 0; i < k; i++)
        result *= n;
    return result;
}
void main(){
    raiseIntToPowerTest();
}
```

```
void raiseIntToPowerTest(){
    printf("%d\n",raiseIntToPower(3,1));//3
    printf("%d\n",raiseIntToPower(3,4));//81
    printf("%d\n",raiseIntToPower(3,3));//27
    printf("%d\n",raiseIntToPower(-3,5));//-243
    printf("%d\n",raiseIntToPower(0,4));//0
    printf("%d\n",raiseIntToPower(0,0));//1
    printf("%d\n",raiseIntToPower(5,0));//1
}
```

Réécrire cette fonction pour qu'elle fonctionne de manière récursive, tout en profitant des informations suivantes :
- Si k est pair, n^k est le carré de n élevé à la puissance $k/2$.
- Si k est impair, n^k est le carré de n élevé à la puissance $k/2$ multiplié par n.

Pour résoudre ce problème, vous devez identifier les cas simples nécessaires pour compléter la définition récursive. Vous devez également vous assurer que votre code est efficace dans le sens qu'il ne fait qu'un appel récursif par niveau de la décomposition récursive. Dans votre code, vous ne pouvez déclarer aucune variable locale de n'importe quel type, utilisez uniquement les paramètres.

Question 3 : Liste chaînée de dominos (*10 pts*)

Le jeu des dominos est joué avec des pièces rectangulaires composées de deux carrés reliés, chacun étant marqué d'un certain nombre de points. Par exemple, chacun des cinq rectangles suivants représente un domino :

Les dominos peuvent être connectés de bout en bout pour former des chaînes, sous réserve que deux dominos ne puissent être reliés ensemble que si les nombres correspondent. Par exemple, vous pouvez former une chaîne composée des cinq dominos en les connectant dans l'ordre suivant (remarquez que nous avons fait pivoter quelques pièces) :

1. Définir le type `noeud`, représentant un domino dans une liste, ayant deux champs de type entier `gauche` et `droite` et un champ `suivant`.
2. Écrire la fonction `rotate180` qui étant donné un nœud, fait pivoter le domino qu'il représente de 180 degrés de sorte que les deux côtés gauche et droite commutent.
3. Ecrire la fonction `formsDominoChain` qui étant donné une liste de dominos renvoie 1 si les nœuds de la liste forment une chaîne de domino, 0 sinon. La fonction peut faire pivoter les nœuds pour pouvoir les ajuster dans la chaîne.

Bonne chance !!