AP1 TP

12 Semaine 12 : Récursivité 1/1

Pour chaque exercice, il est demandé d'écrire un programme qui permet d'examiner la trace du programme.

Exercice 1 : Puissance

1. Soient a et n deux entiers positifs. Écrire un algorithme qui calcule a^n.
2. Quelle est la complexité de l'algorithme proposé ?
3. Ecrire un algorithme calculant la puissance reposant sur l'observation suivante :
   si n est pair, $a^n = a^{n/2} \cdot a^{n/2}$; si n est impair, $a^n = a \cdot a^{(n-1)/2} \cdot a^{(n-1)/2}$.
4. Quelle est la complexité de ce nouvel algorithme.

Tests	Résultat(s) attendu(s)	Résultat(s) observé(s)
a=2 ;n=3		
a=-2 ;n=3		
a=2 ;n=-3		
a=2 ;n=0		
a=0 ;n=3		
a=2 ;n=30000		

Exercice 2 : Suite de Fibonacci

Ecrire un algorithme récursif calculant le n-ième terme de la suite de Fibonacci : $u_1 = 1, u_2 = 1, u_n = u_{n-1} + u_{n-2}$.

Tests	Résultat(s) attendu(s)	Résultat(s) observé(s)
n=5 ;		
n=2 ;		
n=1 ;		
n=0 ;		
n=-2 ;		
n=3000000 ;		

Exercice 3 : Fonction d'Ackerman

La fonction d'Ackerman $A : (m, n) \to A(m, n)$ est définie sur $N \times N$ par :

Si $m = 0$, alors $A(0, n)$ = $n + 1$, sinon si $n = 0$, $A(m, 0) = A(m - 1, 1)$, sinon $A(m, n) = A(m - 1, A(m, n - 1))$.

Ecrire un algorithme récursif calculant $A(m, n)$.

Tests	Résultat(s) attendu(s)	Résultat(s) observé(s)
$A(0,0)$;		
$A(0,1)$;		
$A(2,3)$;		
$A(-2,-3)$;		

Exercice 4 : Horner et polynôme

Le schéma de Horner pour calculer la valeur en x d'un polynome de degré n

$$a_0 + a_1x + a_2x^2 + ... + a_nx^n$$

est de la forme :

$$(...((((a_n) * x + a_{n-1}) * x + a_{n-2})...) * x + a_1) * x + a_0$$

Ecrivez une fonction récursive renvoyant la valeur du polynôme reçu pour la valeur de x donnée.

Tests	Résultat(s) attendu(s)	Résultat(s) observé(s)
$a_{0...n} = 1$ x=0 n=5		
$a_{0...n} = 1$ x=-1 n=5		
$a_{0...n} = 1$ x=2 ; n=5		
$a_{0...n} = 1$ x=2 ; n=0		
$a_{0...n} = 1$ x=2 ; n=-2		

Exercice 5 : Ping-Pong

Soient les deux fonctions suivantes :

Action *Ping* (n : entier) : entier
Début
  Si $n = 0$ **Alors**
    Afficher ("Point Ping") ;
  Sinon
    Pong($n - 1$) ;
    Afficher("Ping ") ;
Fin

Action *Pong* (n : entier) : entier
Début
  Si $n = 0$ **Alors**
    Afficher ("Point Pong") ;
  Sinon
    Ping($n - 1$) ;
    Afficher("Pong ") ;
Fin

Implémenter.

Exercice 6 : Hanoï

Le probème des tours de Hanoï est un jeu de réflexion imaginé par le mathématicien français Edouard Lucas, et consistant à déplacer des disques de diamètres différents d'une tour de "départ" à une tour d' "arrivée" en passant par une tour "intermédiaire" et ceci en un minimum de coups, tout en respectant les règles suivantes :

Tests	Résultat(s) attendu(s)	Résultat(s) observé(s)
n=0 n=-2 n=1 n=2 n=3		

1. on ne peut pas déplacer plus d'un disque à la fois,
2. on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

Nous disposons de 3 axes A, B, C. En position de départ, l'ensemble des disques (n au total) sont empilés par taille décroissante sur l'axe A. Les axes B et C sont vides.

Ecrire une foncion récursive qui permet de déplacer les n disques de l'axe A vers l'axe C (en passant par l'axe B) en respectant les règles du jeu.