Algorithmique 11 : récursivité

Alain Legendre (avec Alain Satabin et Nicolas François)

Exercice 1 : somme des premiers entiers

Une fonction (ou plus, généralement un algorithme) qui contient un appel à elle-même est dite récursive. Un exemple très simple est donné par la somme $S_n = 1 + 2 + \ldots + n$ des n premiers entiers. La relation évidente $S_n = S_{n-1} + n$ montre que, pour calculer S_n, il suffit de calculer S_{n-1}, et donc de faire appel à la même fonction « somme » au rang précédent.

Rédiger un algorithme récursif de calcul de la somme des n premiers entiers, où $n \geq 1$ est fourni par l'utilisateur.

Exercice 2 : calcul d'une puissance

En remarquant que $x^n = x \cdot x^{n-1}$, rédiger un algorithme récursif de calcul de x^n, où x et n sont fournis par l'utilisateur.

Exercice 3 : factorielle

On rappelle que la factorielle d'un entier n est l'entier noté $n!$ défini par $0! = 1$ et $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$ pour tout $n \geq 1$.
La relation évidente $n! = n \cdot (n-1)!$ (pour tout $n \geq 1$) montre que pour calculer $n!$, il suffit de calculer $(n-1)!$, et donc de faire appel à la même fonction « factorielle » au rang précédent. Et pour les valeurs $n = 0$ ou $n = 1$, la fonction factorielle donnera directement le résultat égal à 1.

Rédiger un algorithme récursif de calcul de la factorielle d'un entier n entré par l'utilisateur.

Exercice 4 : suite de Fibonacci

On rappelle que la suite de Fibonacci est la suite numérique (f_n) définie par $f_0 = f_1 = 1$ et $f_n = f_{n-1} + f_{n-2}$ pour tout entier $n \geq 2$. Ainsi, pour calculer un terme, il suffit de calcuelr les deux termes précédents. Rédiger un algorithme récursif de calcul du n-ième terme de la suite de Fibonacci, pour un entier n fourni par l'utilisateur.

Exercice 5 : tours de Hanoï

Le problème des tours de Hanoï est un jeu imaginé par Edouard Lucas[1], consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », comme l'illustre la figure ci-dessus.

1. mathématicien français, 1842-1891, né à Amiens.

Le résultat doit être obtenu en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer qu'un seul disque à la fois,
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide (et on suppose que cette dernière règle est également respectée dans la configuration de départ).

Après quelques essais sur un petit nombre de disques, on voit que cet objectif peut être réalisé en utilisant un algorithme récursif simple. Ecrire alors une fonction affichant chaque déplacement de disque pour un nombre initial n de diques, fourni par l'utilisateur.

Exercice 6 : fonction d'Ackermann

La fonction d'Ackermann [2] est une fonction $A(m,n)$ de deux variables entières prenant des valeurs très importantes, même pour m et n assez « petits » (par exemple, $A(4,2) = 2^{65536} - 1$.. !).
Formellement, cette fonction est elle-même définie de façon récursive par

$$A(m,n) = \begin{cases} n+1 \text{ si } m = 0 \\ A(m-1,1) \text{ si } m > 0 \text{ et } n = 0 \\ A(m-1, A(m,n-1)) \text{ si } m > 0 \text{ et } n > 0 \end{cases}$$

- A titre d'observation, calculer directement les nombres $A(m,n)$ pour $m = 0,1,2,3$ et $n = 0,1,2,3,4$.
- Rédiger un algorithme récursif de calcul de $A(m,n)$ pour (m,n) fourni par l'utilisateur.

2. Wilhem Ackermann, 1896-1962, mathématicien allemand, connu pour ses travaux en théorie de la programmation.