Informatique – TP n° 1 – Révisions

Exercice 1 – Soit la suite définie par $u_0 = 0$, et pour tout $n \in \mathbb{N}$, $u_{n+1} = \sqrt{3u_n + 4}$. On montre facilement que cette suite converge vers 4 en croissant.

1. Écrire un programme demandant à l'utilisateur un entier n en renvoyant tous les termes de la suite jusqu'à u_n .
2. Écrire un programme renvoyant le plus petit entier n pour lequel $u_n > 3,99999999$.

Exercice 2 – Calculer $\sum_{n=0}^{1000} u_n$ où $u_0 = 1$ et $\forall n \geq 0$, $u_{n+1} = \frac{1}{u_n + 1}$.

Exercice 3 – (sans machine)

Soit $F_0 = 0$, $F_1 = 1$ et pour tout $n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$ (suite de Fibonacci). Écrire une fonction prenant en paramètre une valeur de n et calculant F_n .

Exercice 4 – On définit la suite de Syracuse par $u_0 \in \mathbb{N}^*$, et

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

On veut vérifier la propriété suivante : il existe un rang N tel que $u_N = 1$ (et à partir de ce rang, la suite boucle : 4, 2, 1, 4, 2, 1, etc.). Écrire un programme demandant à l'utilisateur une valeur initiale u_0 , calculant et affichant les différents termes de la suite tant qu'ils ne sont pas égaux à 1, et affichant pour terminer la première valeur de N pour laquelle $u_N = 1$, ainsi que la plus grande valeur obtenue pour u_n .

Cette propriété, à l'énoncé pourtant très simple, est encore aujourd'hui une conjecture, malgré les efforts acharnés de nombreux mathématiciens plus brillants les uns que les autres.

Exercice 5 – Écrire une fonction prenant en paramètre deux réels strictement positifs a et e , et calculant une valeur approchée de la somme (dont on justifiera la convergence) $\sum_{n=1}^{+\infty} \frac{(-1)^n}{n^a}$ à e près.

Exercice 6 – Soit $f(x, y) = x \cos y + y \cos x$. On définit une suite u_n par $u_0 = 0$, $u_1 = 1$, $u_{n+2} = f(u_{n+1}, u_n)$ si n est pair, et $u_{n+2} = f(u_{n+1}, u_{n-1})$ si n est impair. Afficher les N premières valeurs de (u_n) . Modifier le programme pour répondre à la même question avec $f(x, y) = xe^y - ye^x$.

Exercice 7 – (sans machine)

1. Écrire une fonction déterminant le maximum des valeurs d'un tableau passé en paramètre
2. Écrire une procédure prenant en paramètre un tableau et triant ses éléments. On procédera par un tri « par insertion » consistant à prendre les éléments les uns après les autres et à les placer à la bonne place parmi les éléments déjà triés.
3. Écrire une procédure déterminant la médiane d'un tableau pris en paramètre.

Exercice 8 – Écrire une procédure prenant en entrée un tableau contenant dans ses n premières entrées les $n + 1$ coefficients binomiaux $\binom{n}{k}$, $k \in \{0, \dots, n\}$, (n -ième ligne du triangle de Pascal), et calculant la ligne suivante (utiliser le même tableau). Écrire un programme affichant sous forme de triangle le triangle de Pascal jusqu'à la ligne 20

Exercice 9 – Écrire une procédure affichant tous les entiers premiers inférieurs à 32000 (on procédera à l'aide d'un crible d'Eratosthène).

Exercice 10 – (sans machine)

On représente un polynôme $a_d X^d + \dots + a_1 X + a_0$ sous forme d'un tableau $[a_0, \dots, a_d]$. Comme on ne peut pas définir une taille variable pour un tableau, on travaille avec des polynômes de degré au plus N , N étant une constante (par exemple 50). Un polynôme est alors représenté par un tel tableau et son degré (ou une borne de son degré), afin d'éviter des calculs inutiles. Contrairement à la convention mathématique, du fait qu'on ne peut pas manipuler des infinis en informatique, on adoptera la convention suivante : le polynôme nul sera de degré -1 .

Écrire des procédures ou fonctions faisant les choses suivantes :

1. Étant donné un polynôme P et une valeur a , pris en paramètre, calcul de $P(a)$ par l'algorithme de Hörner.
2. Étant donné P et son degré d , calcul de P' et de son degré, dans les mêmes variables.
3. Étant donné un polynôme P , et un réel a , calcul de la multiplicité de a comme racine de P (éventuellement 0).