

Série de TP n° = 3 Programmation.

Exercice n° = 1 : Soit la suite u_n définie par

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

Il existe une conjecture non démontrée à propos de cette suite qui s'appelle conjecture de Collatz.

Conjecture 1 $\forall u_0 \in \mathbb{N}$ la suite tend vers la période 4,2,1.

On souhaite tester de façon expérimentale cette conjecture, écrire une procédure qui permet de lire le terme initial u_0 et le nombre de termes n à tester et de tester la conjecture de Collatz.

Exercice n° = 2: Soient les procédures définies par :

fact1:=proc(n)	fact2:=proc(n)	fact3:=proc(n::integer)
local i,factn;	local i,factn;	local i,factn;
factn:=1;	factn:=n;	factn:=1;
for i from 1 to n do	i:=n;	for i from 1 to n do
factn:=factn*i;	while i > 1 do	factn:=factn*i;
od;	i:=i-1;	od;
end;	factn:=factn*(i);	end;
	od;	
	end;	

Sans faire de tests sur machine, dire ce qui se passe lors de l'exécution des appels suivants :

fact1(5)	fact2(5)	fact3(5)	fact1(1/2)	fact2(1/2)	fact3(1/2)
fact1(5.)	fact2(5.)	fact3(5.)	fact1(0)	fact2(0)	fact3(0)

Exercice n° =3: On souhaite créer une liste L sous la forme suivante :

$$L = \left[\pi, \underbrace{\frac{e^{-\pi^2}}{\pi}, \frac{e^{-\pi^2}}{\pi}, \dots, \frac{e^{-\pi^2}}{\pi}}_{n-1 \text{ fois}} \right]$$

Limiter l'arithmétique à 5 chiffres significatifs en posant Digits:=5:

1. Créer une procédure qui permet d'effectuer les tâches suivantes :
 - Lire n et calculer la liste L sous forme de nombres exprimés en virgule flottante. (Terminer la commande par : pour éviter l'affichage).
 - Calculer à l'aide d'une boucle for la somme $L[1] + L[2] + \dots + L[n]$.
2. Même question que (1) mais pour la somme $L[n] + \dots + L[2] + L[1]$.
3. Tester les deux procédures pour $n = 2 \times 10^5$ et comparer les résultats.

Exercice n° = 4 : Expliquer les réponses du logiciel dans les cas suivants: Expliquer les réponses du logiciel dans les cas suivants:

<pre>F := proc (L::list) local LL; LL := seq(args(i), i = 1 .. nargs); op(1, F) := LL end proc; F([1, 2, 3]); Error, (in F) invalid left hand side in assignment</pre>	<pre>W := proc (n, k) local i, T; i := 1; T := 1; while i <= n do T := T+i^k; i := i+2 end do; print(T) end proc; ln(W(5, 2)); 36 Error, (in ln) expecting 1 argument, got 0</pre>
<pre>SumList := proc (L::list) local S, i; for i to nops(L) do S := S+L[i] end do end proc; SumList([]) "Le logiciel n'affiche rien"</pre>	<pre>Explist := proc (L) local i; for i to nops(L) do if 0 < L[i] then L[i] := exp(L[i]) elif L[i] < 0 then L[i] := exp(-L[i]) end if end do end proc; L := [-5, 3, 0, 1, -2]; Explist(L); Error, (in Explist) illegal use of a formal parameter</pre>