Episode 2 : boucles, tests, programmes

1 Premieres fonctions

EXERCICE 1 Voici le code de deux fonctions.

```
fibo_rec:=n->if n<=1 then n else fibo_rec(n-1)+fibo_rec(n-2) fi;
fibo_it:=proc(n)
  local a,b,i;
  a:=0:b:=1;
  if n<=1 then RETURN(n) fi;
  for i from 2 to n
   do  c:=a+b; a:=b; b:=c  od;
  RETURN(b)
end;
```

1. Que retournent `fibo_rec(3)` et `fibo_it(3)` ?
2. Ecrire une fonction `factorielle` prenant en entrée $n \in \mathbb{N}$ et retournant $n!$
   On écrira nue version itérative, et une version récursive.

EXERCICE 2
1. Construire une liste $[1, 2, \dots, 50]$ constituée des 50 premiers entiers > 0. On utilisera successivement `seq` et une boucle `for ... do ... od`.
2. Construire la liste des 50 premiers entiers premiers, puis la liste des entiers ≤ 500 qui sont premiers. Combien en trouve t-on ?

EXERCICE 3 Ecrire une procédure `maxi` prenant en entrée une liste de réels et retournant le maximum de ces réels. Par exemple, `maxi([1,8,-5])` ; retournera 8 comme résultat.

EXERCICE 4 Ecrire une procédure `appartient` qui prend deux arguments x et L, et qui retourne `true` ou `false`, selon que x est ou non l'un des termes de la liste L.

EXERCICE 5 Ecrire une procédure `doublon` prenant en entrée une liste, et retournant `true` ou `false`, selon qu'il y a ou non une valeur doublée dans la liste (un terme apparaissant deux fois ou plus).

EXERCICE 6 (*) Ecrire des procédures prenant en entrée une liste de réels, et déterminant si les éléments de la liste forment une suite :
- croissante ;
- monotone ;
- de signe (large) constant.

EXERCICE 7 Ecrire une procédure `inversion` prenant en argument une liste, et retournant cette même liste en ayant inversé l'ordre des éléments.

EXERCICE 8 (DS 98) Un polynôme $P = 3 + 7X^2 - 4X^3 + X^5$ peut être représenté par la liste de ses coefficients (en commençant par le coefficient constant : ici `[3,0,7,-4,0,1]`). La méthode de Hörner consiste à l'évaluer en n sommes et n produits (où n est le degré du polynôme) :

$$P(t) = 3 + t\Big(0 + t\Big(7 + t\big(-4 + t(0 + t.1)\big)\Big)\Big).$$

Ecrire une procédure `hörner` prenant en entrée un polynôme (sous la forme de la liste de ses coefficients) et un réel (ou une variable) x, et retournant l'évaluation de ce polynôme en ce réel grâce à la méthode de Hörner. Par exemple, `hörner([1,-1,2,3],-2)` retournera -13, alors que `hörner([1,1,0,1],toto)` retournera $1 + toto + toto^3$.

2 Programmes plus élaborés

EXERCICE 9 Ecrire une procédure `ramanujan`[1] prenant en entrée un entier N, et retournant la suite (ou liste...) des entiers s'écrivant de deux façons différentes comme somme de deux cubes d'entiers $\leq N$. Par exemple, $1729 = 12^3 + 1^3 = 10^3 + 9^3$ est le premier entier concerné. Il doit apparaître dans `ramanujan(12)` mais pas `ramanujan(10)`.

On pourra chercher les égalités $a^3 + b^3 = c^3 + d^3$ avec a le plus petit de a, b, c, d, et $c \leq d$. Cela reviendra donc à faire 4 boucles imbriquées.

EXERCICE 10 Montrer que pour $n \in [\![1, 100]\!]$, la dérivée n-ième de $x \mapsto x^{n-1}\mathrm{e}^{-1/x}$ est égale à $\dfrac{\mathrm{e}^{-1/x}}{x^{n+1}}\cdot$

EXERCICE 11 Montrez que les 100 premiers polynômes cyclotomiques[2] ont leurs coefficients dans $\{-1, 0, \}$. Le résultat est-il valable au delà ?

EXERCICE 12 Calculer $f \circ f \circ f(4444^{4444})$, où f est l'application qui associe à un entier la somme des chiffres de sa représentation en base 10. Question subsidiaire : retrouver le résultat à la main !

EXERCICE 13 Ecrire une procédure `TVI` simulant la dichotomie vue dans la preuve du théorème des valeurs intermédiaires : la procédure prend une fonction f (continue), deux réels a et b tels que $f(a)f(b) < 0$, un réel $\varepsilon > 0$, et renvoie $c \in [a, b]$ tel que $f(x) = 0$ pour un certain $t \in [c - \varepsilon, c + \varepsilon]$. Exemples :

```
> TVI(x->x^2-2,1,2,.0001),TVI(sin,3,4,.000001);
                          1.414245606, 3.141592983
> evalf(sqrt(2)),evalf(Pi);
                          1.414213562, 3.141592654
```

EXERCICE 14 Donner la liste des solutions du problème suivant : on cherche à remplir les 8 cases vides du tableau suivant, sachant que les entiers recherchés sont tous distincts et dans $[\![1, 16]\!]$:

$$\begin{array}{ccccccc}
\cdot & + & \cdot & - & \cdot & = & 12 \\
+ & & + & & + & & \\
17 & + & \cdot & - & \cdot & = & 16 \\
- & & - & & - & & \\
\cdot & + & \cdot & - & \cdot & = & 11 \\
= & & = & & = & & \\
18 & & 8 & & 9 & &
\end{array}$$

Une solution sera sous la forme d'une liste de 9 entiers, comme par exemple : `[2,13,3,17,9,10,1,14,4]`. Combien trouvez-vous de solutions ?

On a (fortement) intérèt à chercher à résoudre le problème la main, pour déterminer l'algorithme que l'on va programmer.

EXERCICE 15 Ecrire une procédure renvoyant le nombre de zéros terminaux dans la représentation décimale de $n!$

Attention, la procédure ne doit bien entendu pas calculer $n!$...

EXERCICE 16 On définit par récurrence les *nombres de Catalan* : $c_0 = 1$, puis $c_n = \displaystyle\sum_{k=0}^{n-1} c_k c_{n-1-k}$ si $n \in \mathbb{N}^*$, de sorte que $c_1 = 1$, $c_2 = 2$ et $c_3 = 5$.

1. Ecrire une procédure `catalan1` **récursive** prenant en argument un entier n, et retournant c_n. Quel est le problème de cette procédure ?
2. Ecrire une procédure **non récursive** `catalan2` faisant le même travail, en créant, pour le calcul de c_n, un tableau de taille n permettant de stocker petit à petit les caleurs des c_k.

[1] La très belle origine du nom de cette procédure vous sera expliquée en TD...

[2] Vous ne savez pas ce qu'est un polynôme cyclotomique ? Maple oui !