Initiation à un logiciel de calcul formel

L'objectif de cette séance est de dégrossir l'utilisation d'un logiciel de calcul formel, ici Xcas, logiciel assez proche (mais assez proche seulement !) de Maple.
Pour débuter un programme (ou une procédure) sur Xcas, aller dans `Prg` puis choisir `Nouveau programme` ou faire directement `alt+P`. Une fois le programme écrit, `f9` permet de le compiler pour pouvoir l'utiliser dans une ligne de commande. En cas de problème ou pour voir comment fonctionne le programme, on peut utiliser la fonction `debug(nom du programme(arguments))` puis `f5` pour exécuter le programme pas à pas (on peut voir notamment la valeur prise par les différentes variables).
Vous pouvez ouvrir une session par exercice et les enregistrer. A chaque ouverture d'une session bien penser à compiler les programmes déjà écrits. Pour débuter, vous pouvez utiliser la commande `ajouter` qui propose des instructions pré-remplies.

ATTENTION : L'indexation d'une liste, d'un vecteur, ou d'une matrice de taille n se fait de 0 à $n-1$.

Exercice 1 *Voici une série de programmes. Dans chaque cas, donner le ou les arguments, les variables utilisées, le ou les résultats du programme.*

```
test1():={
 local k;
 for (k:=0;k<=pi/2;k:=k+pi/10){
   print(sin(k))
 }
}:;

test2(n):={
 local k,(f:=1);
 for(k:=1;k<=n;k++){
  f:=f*k
 }
 return f
}:;

test3(a,b):={
 local r;
 while (b!=0) {
   r:=irem(a,b);
   a:=b;
   b:=r;
 }
 return(a);
}:;
```

Exercice 2 *On souhaite approcher la constante γ d'Euler par la formule*

$$\gamma = \lim_{n\to\infty} 1 + \frac{1}{2} + \frac{1}{3} \ldots + \frac{1}{n} - \ln n$$

Pour cela on écrit une procédure qui calcule les 1000 premiers termes de la suite et affiche le résultat pour les multiples de 100 uniquement. On propose donc le programme suivant :

```
gammaeuler():={
local k,s;
 for(k:=1;k<=1000;k++){s:=s+1/k;
   if (k%100==0) {
    print(s-ln(k))
   }
 }
}:;
```

Recopier ce programme, le tester et voir comment l'améliorer.

Exercice 3 [1]

La suite de Syracuse est ainsi définie :

- u_0 *est un entier strictement positif.*
- *Pour tout entier* $n > 0$*, on pose :* $u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$

La conjecture de Syracuse affirme que quel que soit l'entier u_0*, la suite boucle sur le cycle* $(4, 2, 1)$.

1. *Écrire une fonction* f *qui prend comme argument un entier* u *et qui renvoie l'entier calculé à partir de* u*. Calculer* $f(1), f^2(1), f^3(1)$.
2. *Pour* $u_0 := 27$*, calculer* u_{111} *à l'aide de* `@@`
3. *Écrire une procédure* orbite *qui à partir d'un entier* u_0 *donné en argument donne la séquence des* u_n *jusqu'au premier* u_n *valant* 1*. Tester la procédure pour* $u_0 = 15$ *puis* $u_0 = 27$
4. *Écrire une procédure* testorb *qui renvoie la longueur et le maximum de l'orbite d'un entier* u_0 *donné en argument. Tester la procédure pour* $u_0 = 27$.
5. *Écrire une procédure* `scan` *qui à partir de deux entier* $m < n$ *donnés en argument renvoie la séquence* $[k, L_k, M_k]$ *où* L_k *et* M_k *sont la longueur et le maximum de l'orbite de* k*,* $m \leq k \leq n$*. Tester la procédure sur le segment*$[5, 9]$.
6. *A l'aide de la fonction* `listplot`*, écrire une procédure* `draworb(u0)` *qui affiche la ligne polygonale joignant les points* $[k, u_k]$ *jusqu'à* $u_n = 1$*. Tester* `draworb(27)`
7. *Tracer la ligne polygonale joignant les points* $[k, M_k]$ *pour* k *allant de* 1 *à* 150

Exercice 4 *D'après le CAPES 2012 Épreuve 2*

- *Pour un ensemble fini* $\mathcal{F}$ *on note* $card(\mathcal{F})$ *son cardinal.*
- *Pour* $n \in \mathbb{N}$ *tel que* $n > 1$*, on note* $\mathcal{I}_n$ *l'ensemble des éléments inversibles de* $(\mathbb{Z}/n\mathbb{Z}, +, \times)$
- *Pour* $(a, b) \in \mathbb{Z}^2$*, le plus grand commun diviseur de* a *et* b *est noté* $a \wedge b$ *(la fonction Xcas est* `gcd(a,b)`*)*
- *On note* $\overline{a}$ *la classe de* a *dans* $\mathbb{Z}/n\mathbb{Z}$
- *On appelle ordre de* $a \in \mathcal{G}$*,* $\mathcal{G}$ *un groupe fini, et on note* $\omega(a)$ *le plus petit élément de* $\{k \in \mathbb{N}^*/a^k = 1_{\mathcal{G}}\}$
- *On rappelle qu'un groupe* $\mathcal{G}$ *est cyclique si et seulement si il existe* $a \in \mathcal{G}$ *tel que* $\omega(a) = card(\mathcal{G})$

[1] on peut répondre à la plupart des questions avec un tableur

Éléments inversibles de l'anneau $(\mathbb{Z}/n\mathbb{Z},+,\times)$

1. *Soient* $(a \in \mathbb{Z}$ *et* $n \in \mathbb{N}$ *tel que* $n > 1$. *Démontrer que* $\overline{a} \in \mathcal{I}_n$ *si et seulement si* $a \wedge n = 1$
2. *Soit* $n \in \mathbb{N}$ *tel que* $n > 1$. *Montrer que* $(\mathcal{I}_n, \times)$ *est une groupe commutatif.*
3. *Sans justification, énumérer les éléments de* $\mathcal{I}_{10}$ *avec leur ordre.* $\mathcal{I}_{10}$ *est - il cyclique ?*
4. *Même question pour* $\mathcal{I}_{12}$
5. *Pour les algorithmes demandés, on utilisera uniquement les opérations* $+, \times, \wedge$, *c'est à dire* `gcd`, *et la fonction à deux variables* `irem` *où* `irem(a,b)` *donne le reste de la division euclidienne de* a *par* b, a *et* b *entiers non nuls*
   On pourra également utiliser des boucles ou des constructions de type :

   ```
   - for
   - while
   - if...then...else...end
   ```

   a) *Écrire une procédure* `Test( , )` *ayant pour arguments deux entiers* k *et* $n > 1$ *affichant* "1" *si* $\overline{k} \in \mathcal{I}_n$ *et* "0" *sinon.*
   b) *Écrire une procédure* `Card( )` *ayant comme argument une entier* $n > 1$ *affichant le cardinal de* $\mathcal{I}_n$.
   c) *Écrire une procédure* `Ord( , )` *ayant comme arguments deux entiers naturels* k *et* $n > 1$ *affichant la valeur de* $\omega(\overline{k})$, *l'ordre de* k *dans* $(\mathcal{I}_n, \times)$, *si* $k \in \mathcal{I}_n$ *et* `"erreur"` *sinon.*
   d) *Écrire une procédure* `Cycl( )` *ayant comme argument* $n \in \mathbb{N}$, $n > 1$ *et affichant si* $\mathcal{I}_n$ *est cyclique ou non.*
   e) *Écrire une procédure* `Inv( , )` *ayant comme arguments deux entiers non nuls* k *et* $n > 1$ *et affichant l'inverse de* $\overline{k}$ *dans* $\mathbb{Z}/n\mathbb{Z}$ *si* $\overline{k} \in \mathcal{I}_n$

Exercice 5 *Agrégation Interne 2012, Épreuve 2*

1. *Démontrer que pour tout entier naturel non nul* n, *il existe un unique couple d'entiers naturels* (p, k) *vérifiant :*
$$n = 2^p + k \text{ avec } k < 2^p$$
2. *Écrire sans utiliser de fonction "puissance", un algorithme donnant à partir d'un entier* n *non nul, l'unique couple d'entiers naturels* (p, k)
3. *On pose* $k = k_n$, *donner les valeurs de* k_n *pour* n *compris entre* 1 *et* 20

Exercice 6 *Faire attention à l'indexation des matrices !*

1. *A partir d'une matrice supposée complètement régulière (on ne vérifiera pas cette hypothèse dans le programme), écrire un programme* `facLU` *ayant comme argument cette matrice et donnant comme résultat les matrices* L, U *de sa factorisation* LU. *Pour déterminer la dimension de la matrice, on pourra utiliser* `coldim` *et pour les opérations élémentaires sur les lignes* `subsop`.
2. *comparer les résultats de* `facLU(A)` *et de* `lu(A)` *avec la matrice* $A = \begin{pmatrix} 4 & 1 & 3 \\ 5 & 2 & 7 \\ 3 & 6 & 2 \end{pmatrix}$ *où* `lu` *est la factorisation* LU *intégrée de Xcas.*

Exercice 7 *On cherche à étudier la structure algébrique de l'ensemble* $\mathcal{A}$ *des matrices de la forme*

$$M = \begin{pmatrix} x+y+z & -x-y & -x-z \\ -x-z & x+y+z & -x-y \\ -x-y & -x-z & x+y+z \end{pmatrix} \text{ où } (x,y,z) \in \mathbb{R}^3$$

1. *Justifier,rapidement, que $\mathcal{A}$ est un espace vectoriel de dimension finie, en donner une base.*
2. *Écrire une procédure* `decompose(P,B)` *qui prend en argument une matrice P et une famille B de matrices $E1\dots Ed$* ***donnée entre crochets*** *(*`B:=[E1,...,Ed]`*. La famille de matrices est supposée libre et ce point n'est pas à vérifier dans la procédure. Celle-ce renvoie le vecteur des coordonnées de P dans B ou* `erreur` *si c'est impossible.*
   Pour résoudre l'équation matricielle, on pourra utiliser la fonction `linsolve` *qui prend comme argument une liste d'équations et d'inconnues et résout le système linéaire correspondant.* `seq` *permet de créer des listes.*
3. *On demande de montrer que $\mathcal{A}$ possède une structure d'algèbre et d'étudier ses propriétés (associativité, commutativité, élément neutre).*
4. *Mêmes questions pour l'ensemble $\mathcal{Q}$ des matrices de la forme*

$$\begin{pmatrix} \alpha + i\delta & \beta + i\gamma \\ -\beta + i\gamma & \alpha - i\delta \end{pmatrix}$$

References

[1] G Dowek. *Les métamorphoses du calcul.* Le Pommier.

[2] JM Ferrard. *Math et Maple.* Dunod.

[3] S Grognet G Connan. *Guide du calcul avec des logiciels libres.* Dunod.

[4] X Gourdon P Dumas. *Maple, Son bon usage en mathématiques.* Springer.