

Programmer avec Xcas



I. Exercice 1 : Somme des inverses

Voici l'algorithme calculant la somme des inverses des entiers de 1 à n .

```
somme_inverses() := {
local k, n, s;
saisir(n);
s:=0;
pour k de 1 jusque n faire
  s:=s+1/k;
fpour
afficher(evalf(s,3));
};
```



Travail à réaliser :

- 1) Créer avec Xcas le programme ci-contre et le tester.
- 2) Créer avec Xcas un nouveau programme afin d'obtenir le premier rang donnant une somme supérieure à 12.



Analyse du programme :

- **Le nom du programme** (sans espaces) suivi d'une paire de parenthèses (c'est une fonction), de := et d'une paire d'accolades ; le code du programme est tapé entre les deux accolades.

- **Déclaration des variables.**

- **Pour demander l'entrée d'une valeur numérique :**

On tape `saisir(n)` pour que le programme demande à l'utilisateur d'entrer la valeur de la variable n .

- **Pour affecter une valeur à une variable :**

On lui affecte la valeur voulue (le symbole d'affectation est :=).

- **Pour insérer une boucle Pour :**

La syntaxe est :

`pour...de...jusque...faire`

compteur valeur initiale valeur finale

`fpour`

Les instructions à exécuter dans la boucle sont placées entre `pour` et `fpour`.

- **Pour faire afficher un résultat :**

On utilise l'instruction `afficher`.

```
somme_inverses() := {
local k, n, s;
saisir(n);
s:=0;
pour k de 1 jusque n faire
  s:=s+1/k;
fpour
afficher(evalf(s,3));
};
```

L'instruction `evalf` permet d'afficher une approximation décimale de s (à 3 décimales ici). Xcas étant un logiciel de calcul formel renverra autrement la valeur exacte de s sous forme de fraction illisible dès que n est grand. Il y a de plus un risque d'erreur ou de plantage du logiciel lorsque l'on travaille sur de grands entiers en valeur exacte.



Outil à utiliser :

- **Pour insérer une boucle Tant Que :**

La syntaxe est :

`tantque...faire`

Test d'arrêt de la boucle :

On utilise : `==` pour est égal à
`!=` pour est différent de

`<` `<=` `>` `>=`

`ftantque`

Les instructions à exécuter dans la boucle sont placées entre `tantque` et `ftantque`.

```
somme_inverses_modif() := {
local s, n;
s:=0;
n:=0;
tantque s<=12 faire
  s:=s+1/n;
  n:=n+1;
ftantque
  afficher(n);
};
```



II. Exercice 2 : Solutions d'une équation du second degré

Voici le début de l'algorithme permettant de déterminer les solutions d'une équation du second degré.

```
second_degre() := {
local a,b,c,D,x1,x2;
saisir(a,b,c);
D:=b^2-4*a*c;
si D<0 alors
  afficher("Pas de solution")
fsi
[
]
};
```



Travail à réaliser :

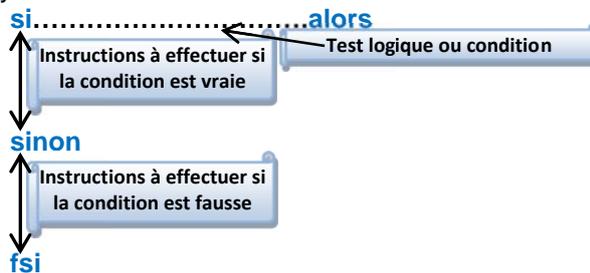
Créer et compléter avec Xcas le programme ci-contre afin d'envisager tous les cas possibles.

Outil à utiliser :



Pour insérer une instruction conditionnelle :

La syntaxe est :



Pour faire afficher un résultat sous forme mathématique :

Si on utilise la fonction **afficher** on obtient :

```
afficher(x1,x2);
fsi
];
];
```

2second_degre()
Deux solutions :
(-5-sqrt(17))/2, (-5+sqrt(17))/2

Si on utilise la fonction **retourne** on obtient :

```
retourne(x1,x2);
fsi
];
];
```

2second_degre()
Deux solutions :
Evaluation time: 4.431
 $(\frac{-5-\sqrt{17}}{2}, \frac{-5+\sqrt{17}}{2})$

Remarque

L'instruction **retourne** en plus de faire afficher le résultat fait sortir d'un programme (fonction) en renvoyant une valeur, permettant ainsi que celle-ci soit utilisée dans un autre programme.

III. Exercice 3 : Le schéma de Bernoulli



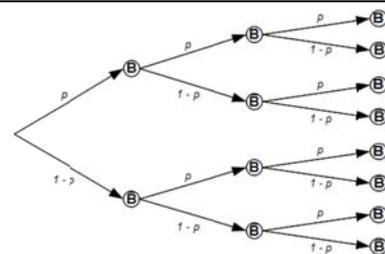
Une urne contient une proportion p de boules blanches ; on effectue n tirages successifs avec remise d'une boule.
On recherche la probabilité d'obtenir k boules blanches lors des n tirages ($0 \leq k \leq n$).



Travail à réaliser :

On veut obtenir par simulation une approximation de cette probabilité.

- Ouvrir le fichier « essais_np.xws ».
- Que fait cet algorithme ?
Comment doit-on l'utiliser ?
- Le tester.
- Modifier ce programme afin d'utiliser une instruction qui affiche le résultat en interrompant le programme et en mettant ce résultat à disposition de tout autre programme.



- Créer un deuxième programme nommé `repete_essais` qui simule 10 000 essais en utilisant le programme précédent et qui affiche les fréquences d'obtention de k boules blanches lors des n tirages ($0 \leq k \leq n$).
- Modifier les valeurs de n et p .

Remarque :

Un programme XCas est une fonction ; il est donc possible de placer des variables dans son nom dont les valeurs seront passées au programme lors de son exécution.

Par exemple :

- on crée un programme qui doit simuler N lancers d'un dé et on le définit par : **lancer_des(N)** ; dans le code du programme N représente le nombre de lancers ;
- pour simuler 500 lancers du dé on tapera ensuite en ligne de commande : **lancer_des(500)**.

Cette méthode évite d'avoir recours à l'instruction **saisir** qui ouvre ensuite une boîte de dialogue demandant la saisie de la valeur de la variable.

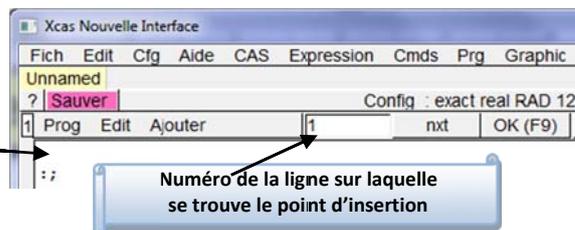
Nouveaux outils à utiliser : 

- **Pour créer un programme :**
Prg → Nouveau programme



On obtient l'interface suivante :

Taper **ici** (ligne 1) le nom du programme (sans espaces) suivi **d'une paire de parenthèses** (c'est une fonction), de **:=** et **d'une paire d'accolades** ; le code du programme est tapé entre les deux accolades.



- **Pour faire afficher les fréquences d'obtention de k boules blanches :**

Pour cela on utilise deux variables de type liste qui permettent de lister le nombre de fois où l'on obtient k boules blanches, et la fréquence correspondante, avec k variant de 0 à n.

Méthode :

On déclare les variables R et F	local R, F;
On initialise les listes vides R et F (ceci a pour effet de donner à R et F le statut de liste)	R:=[]; F:=[];
On définit la taille de la liste et on crée ses différents éléments en les initialisant à 0	pour k de 0 jusque n faire R[k] := 0; F[k] := 0; fpour
On peut ensuite à l'aide d'une boucle pour effectuer les 10 000 simulations et modifier l'élément R[k] à chaque fois que l'on obtient k boules blanches.	
On peut ensuite à l'aide d'une boucle pour affecter à chacun des éléments F[k] de la liste le calcul de la fréquence d'obtention de k boules blanches.	



IV. Exercice 5 : Des points de départ pour des séquences



Travail à réaliser :

- 1) Créer un algorithme permettant de résoudre l'un, au choix, des trois problèmes suivants.
- 2) Coder cet algorithme à l'aide de Xcas

A. Évolution de populations (calcul des termes de suites imbriquées et prévisions)

Dans un pays de population constante égale à 60 millions d'habitants, on compte 20 millions de citadins et 40 millions de ruraux en 2005. Les habitants vivent soit en zone rurale, soit en ville et on constate que les mouvements de population suivent la règle suivante : chaque année, 20% des ruraux émigrent à la ville et 10% des citadins émigrent en zone rurale.

On souhaite calculer le nombre d'habitants dans chaque zone après n années et prévoir l'évolution à long terme.

B. Dichotomie (recherche de solution approchée d'une équation)

Trouver un encadrement de la solution positive de l'équation $x^3 = 3x + 1$.

C. Somme des termes d'une suite récurrente (avec une suite $u_{n+1}=f(u_n ; n)$)

$$\begin{cases} u_0 = 1 \\ \forall n \in \mathbb{N}, u_{n+1} = u_n + 4n + 6 \end{cases}$$

On souhaite calculer $u_0 + u_1 + \dots + u_n$ pour une valeur quelconque de n .



V. Exercice 6 : Le canal du midi



Travail à réaliser :

- 1) Écrire avec XCAS un programme nbsacs(x) qui calcule le nombre de sacs restants à Toulouse en fonction du nombre x de sacs que le marchand avait au départ de Sète.
- 2) Pour rendre le voyage rentable, le marchand souhaite arriver avec au moins la moitié de son chargement de départ. Il cherche la valeur minimale de x pour cela.
Écrire un programme faisant appel au précédent pour répondre à la question.

Au début du 18^{ème} siècle, un marchand veut remonter de Sète jusqu'à Toulouse pour vendre sa farine. Pour cela, il emprunte le canal du Midi qui relie le mer Méditerranée et la Garonne. Ce canal est parsemé de 63 écluses. A chacune d'elles, le marchand doit laisser 1% de son chargement en péage royal, puis échanger 5 sacs de farine contre de la nourriture. L'objectif est de déterminer la quantité de farine qu'il lui reste à vendre à son arrivée à Toulouse.