

## Restez Basiques

Vous rêvez de programmer, mais vous ne voulez pas vous prendre la tête. Vous ne voulez pas vous investir dans l'apprentissage d'un langage compliqué, et vous ne voulez pas non plus investir vos fonds dans un langage de programmation pour une activité annexe. Et bien, malgré toutes ces contraintes, FREELOG peut vous aider. FREELOG et StarOffice.

C'est nouveau ça ?

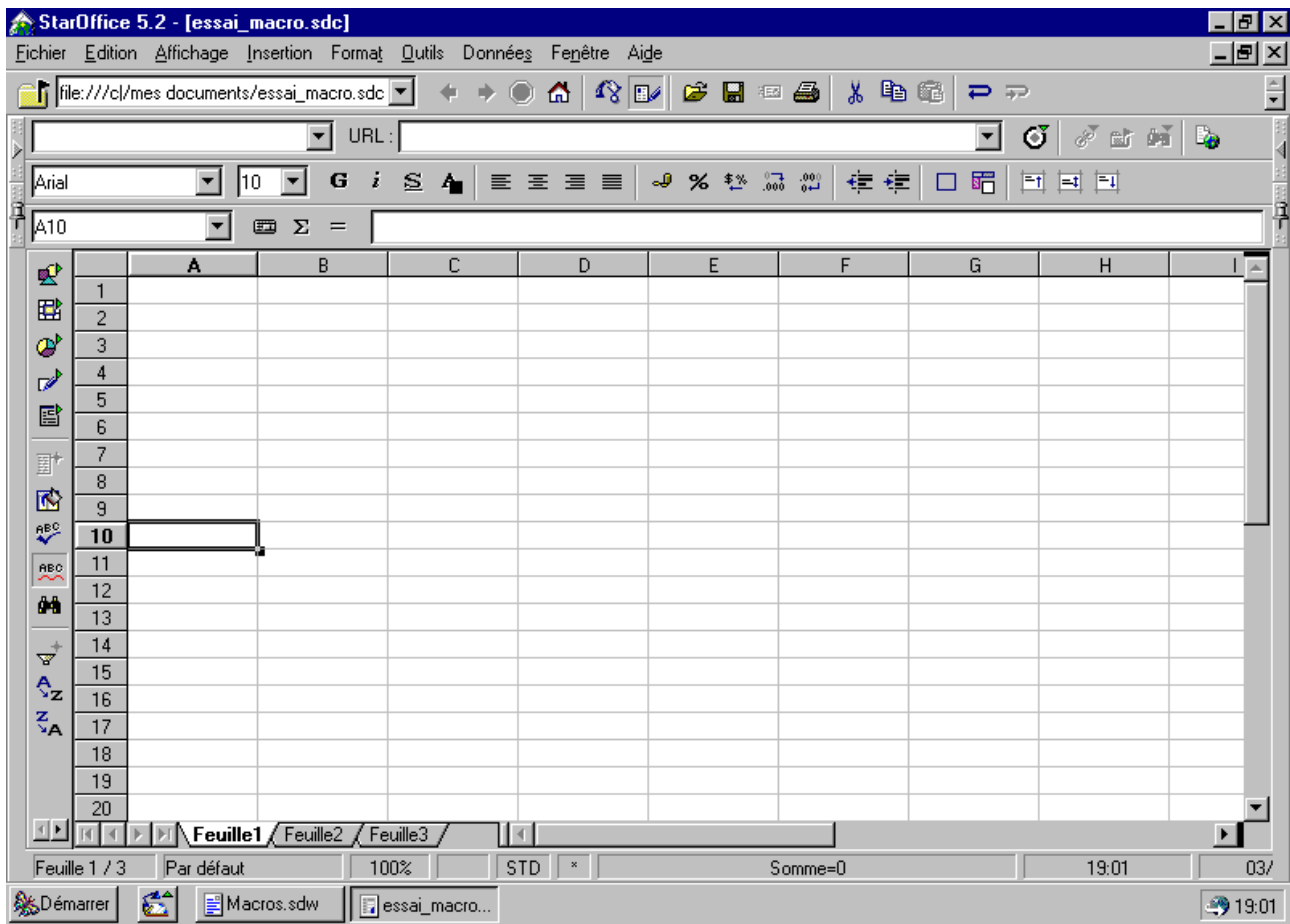
Ben, oui et non. Sans faire un historique complet, lorsque les premiers tableurs sont apparus, les concepteurs de ces programmes se sont vite rendus compte de la nécessité de pouvoir automatiser certaines tâches. Un néologisme fut créé pour désigner ces tâches répétitives : les macros. Embryonnaires au commencement, elles furent bientôt un langage à part entière. Chaque éditeur y allait de son langage macro révolutionnaire. Bref, c'était le règne de la confusion la plus totale, avec des effets d'annonce toujours plus dithyrambiques. L'évolution de chaque langage propriétaire nécessitait de formaliser un peu tout cela. Finalement, le Basic (Beginners All Purpose Symbolic Instruction Code) allait relativement s'imposer comme un standard inter-éditeur, grâce à deux qualités : sa simplicité d'apprentissage et d'utilisation, et ses instructions puissantes.

Où trouver un Basic ?

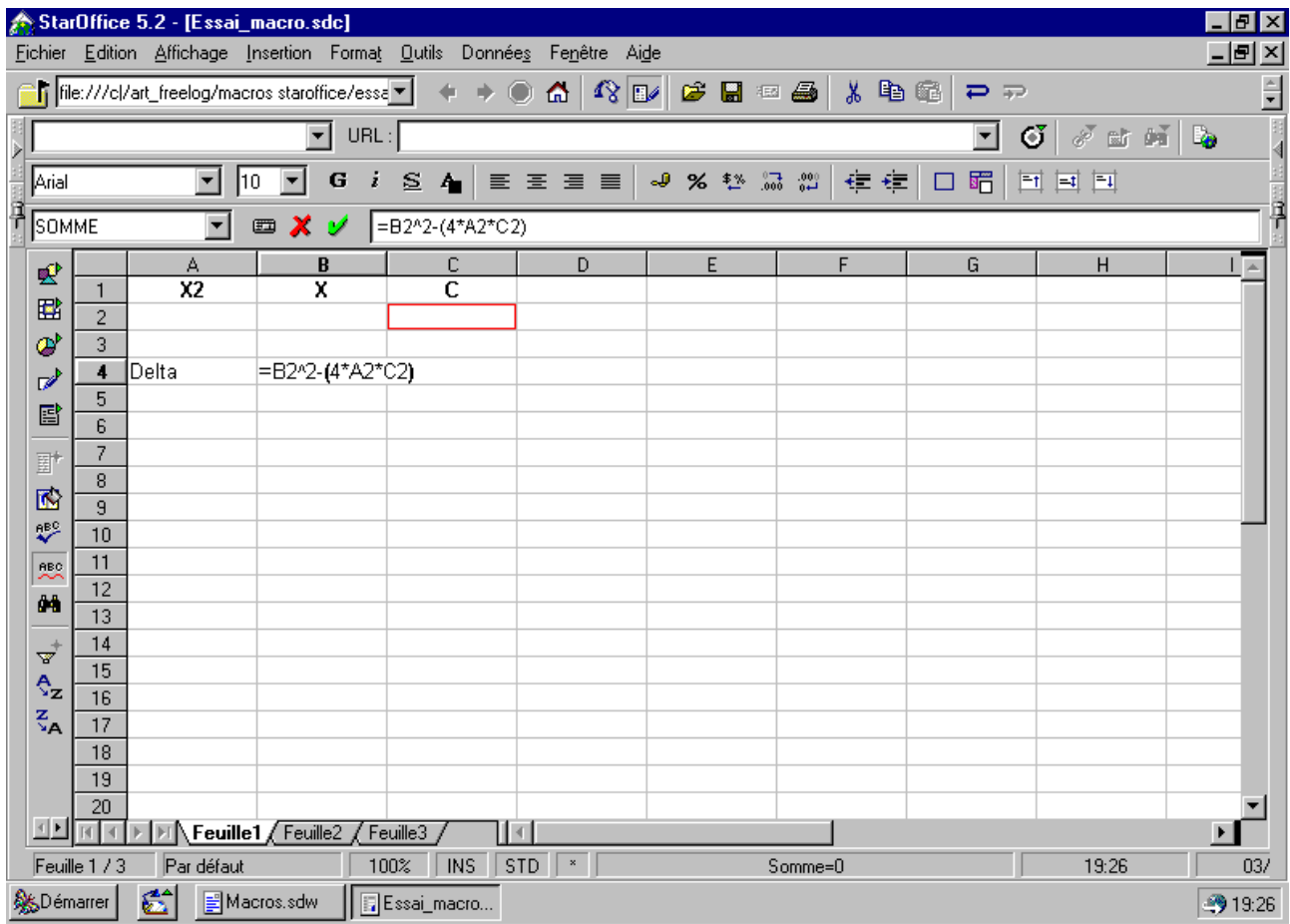
N'importe où ;-). En fait, un basic (le bon vieux GW Basic) est toujours distribué avec les systèmes d'exploitation Windows 9x (si, si... vérifiez, vous l'avez sur votre CD d'installation). Mais il est un peu ancien, donc assez peu convivial. Celui qui nous intéresse est distribué avec StarOffice. Il est tout aussi gratuit; mais plus récent. Autre avantage, les documents produits (feuilles de calcul; courriers...) sont compatibles avec Vous-Savez-Qui ;-). Cette compatibilité est importante. En effet, de nos jours, (quasiment) tout le monde a déjà eu une expérience informatique avec des applications de type tableur ou traitement de textes (ou plus si affinités), mais assez curieusement, les macros ne sont guère utilisées que par moins de 1% des utilisateurs qui se targuent de savoir utiliser un des logiciels précités. Pour ceux d'entre vous qui voudraient aller plus loin avec des produits plus orientés sur la programmation, il existe différents Basics en freeware dont FREELOG est le zéléateur régulier dans ces mêmes colonnes. Quant à nous, nous resterons sur nos positions de départ, avec un développement à partir d'un document de travail. Pour ceux que la chose intéresse, il est aussi possible de développer sous Java. Il est également possible de créer des macros StarOffice, uniquement en utilisant la fonction d'enregistrement, et en indiquant les opérations à faire à la souris. StarOffice est presque aussi performant que son grand frère...

On y va ?

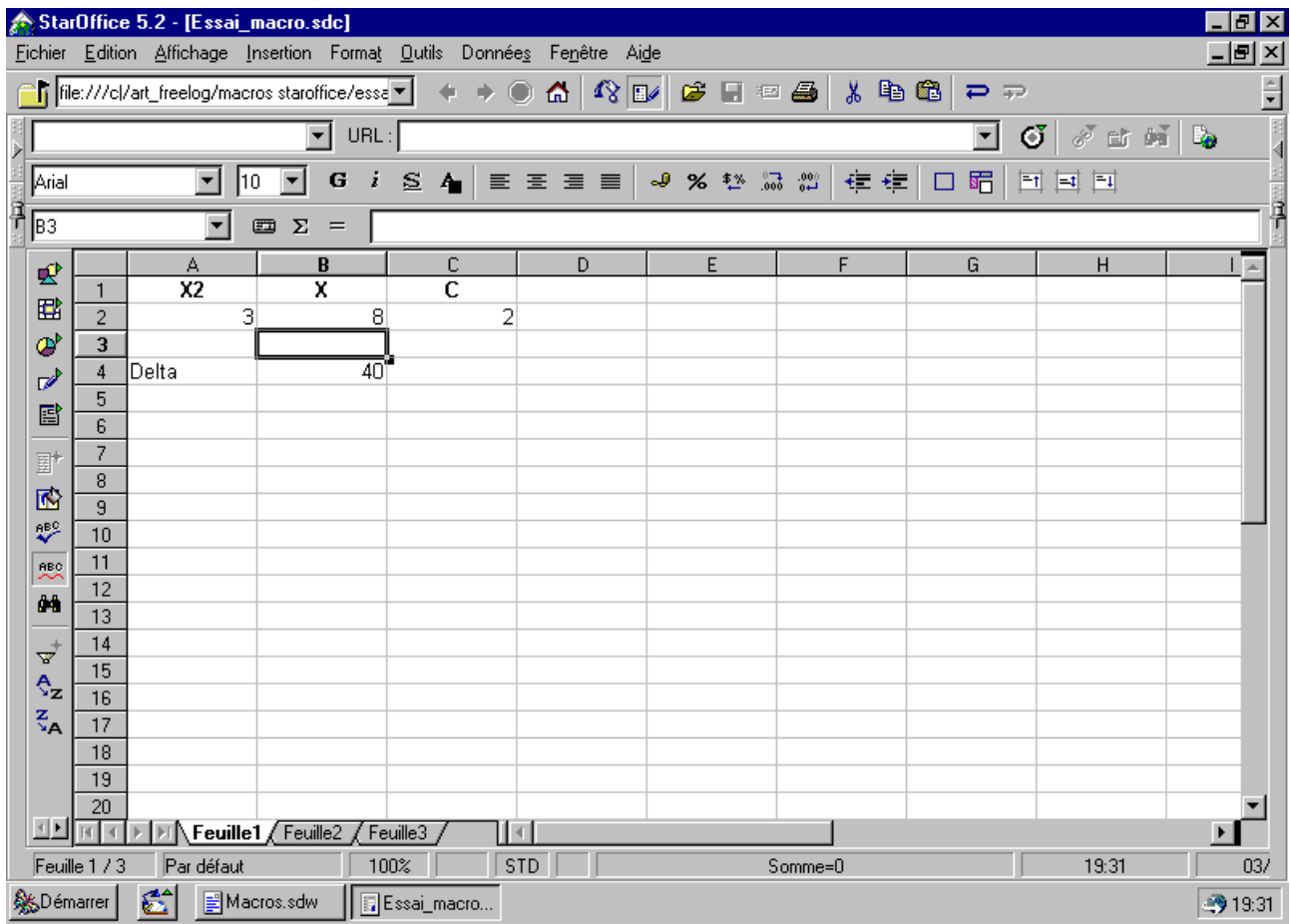
Pour y aller, nous supposons que vous maîtrisez le fonctionnement basique (jeu de mots ;-)) des applications standard sous StarOffice. Notre première application se basera sur une feuille de calcul. Ouvrons donc StarOffice. Voici ce qui nous attend au départ (Fig1) :



Par goût, j'affectionne particulièrement les tableurs. Donc, nous allons utiliser une feuille de calcul. Pour notre première application, soyons modestes. Nous allons faire plaisir aux écoliers pour qui c'est la rentrée. Rappelons-nous nos bons souvenirs sur les discriminants. Nous considérons une équation du seconde degré de la forme  $f(x)=ax^2+bx+c$ . Le discriminant est  $\Delta=b^2-4*a*c$ . Rien de sorcier et nous nous arrêterons là pour l'étude du problème. Voyons comment on peut résoudre cela sur le tableur (Fig 2) :



et voilà le résultat que nous obtiendrons en mettant les valeurs des énoncés directement dans les cases concernées (Fig 2 bis) :

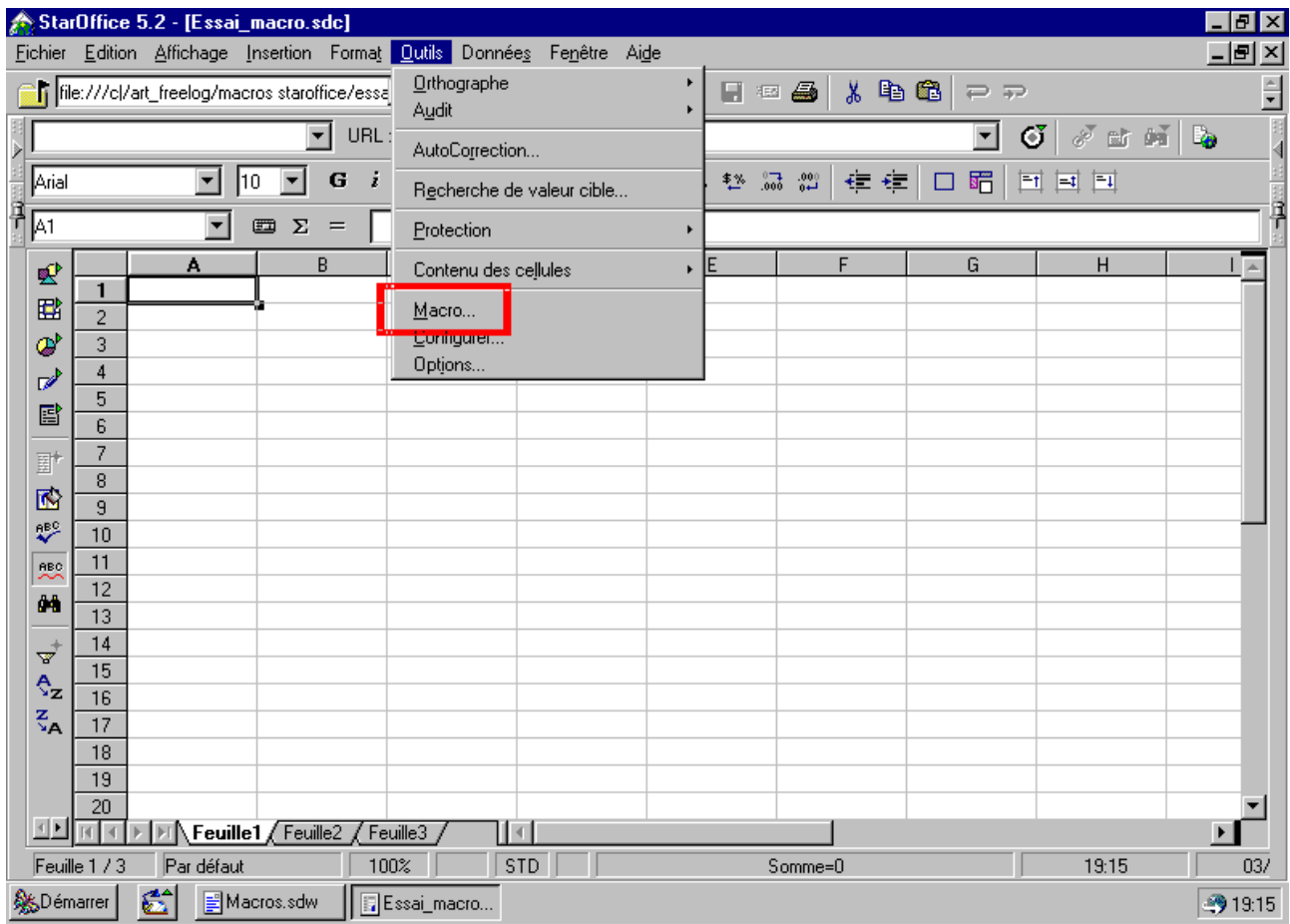


Comme on peut le voir, tout cela ne représente pas une manipulation très compliquée. Essayons maintenant de faire une macro pour réaliser la même opération.

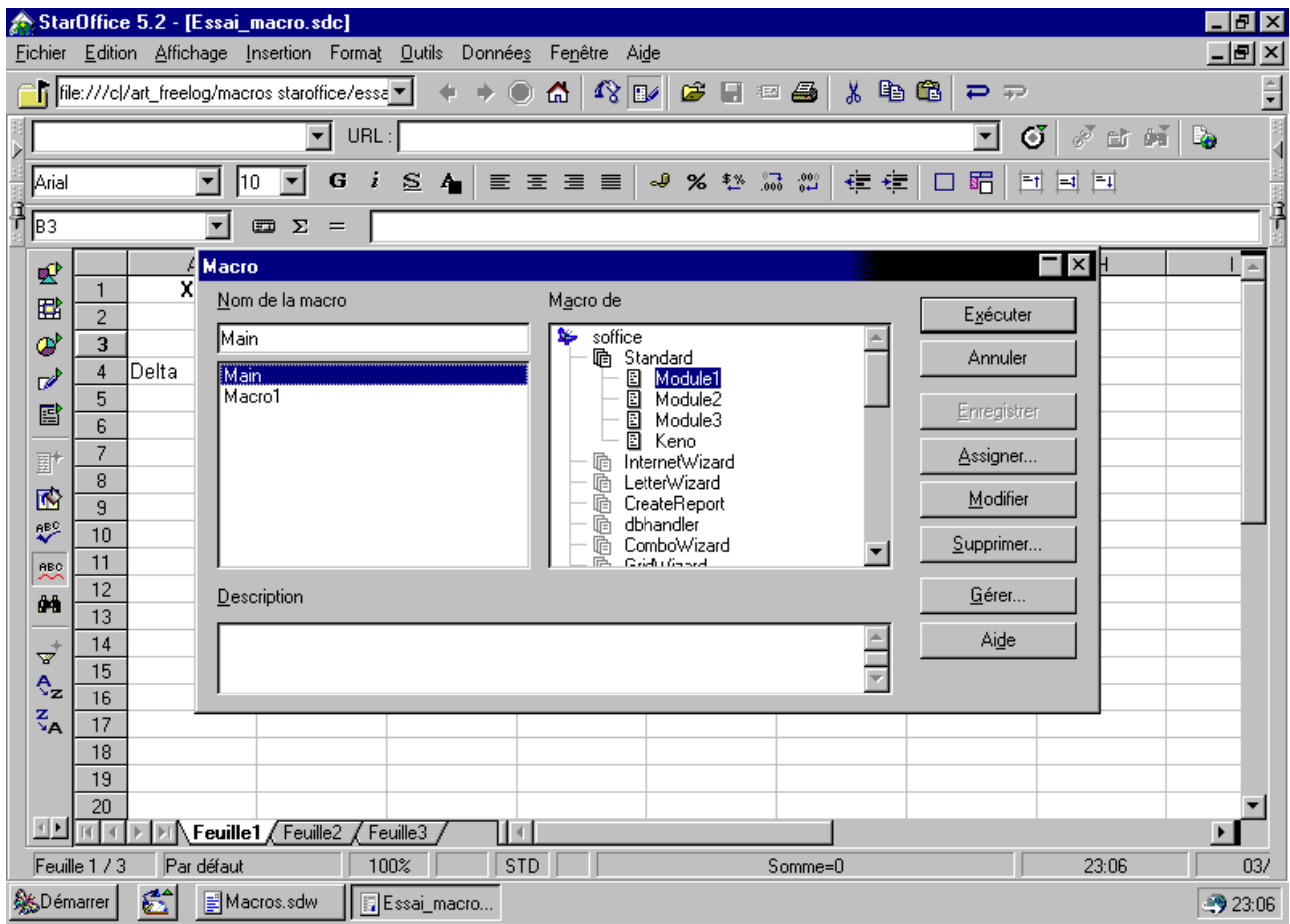
A ce niveau, il est évident qu'une macro est superflue. Mais nous allons voir que certains avantages peuvent apparaître pour des manipulations ultérieures...

Fidèles à nos habitudes, nous allons vous guider pour que votre apprentissage se passe au mieux.

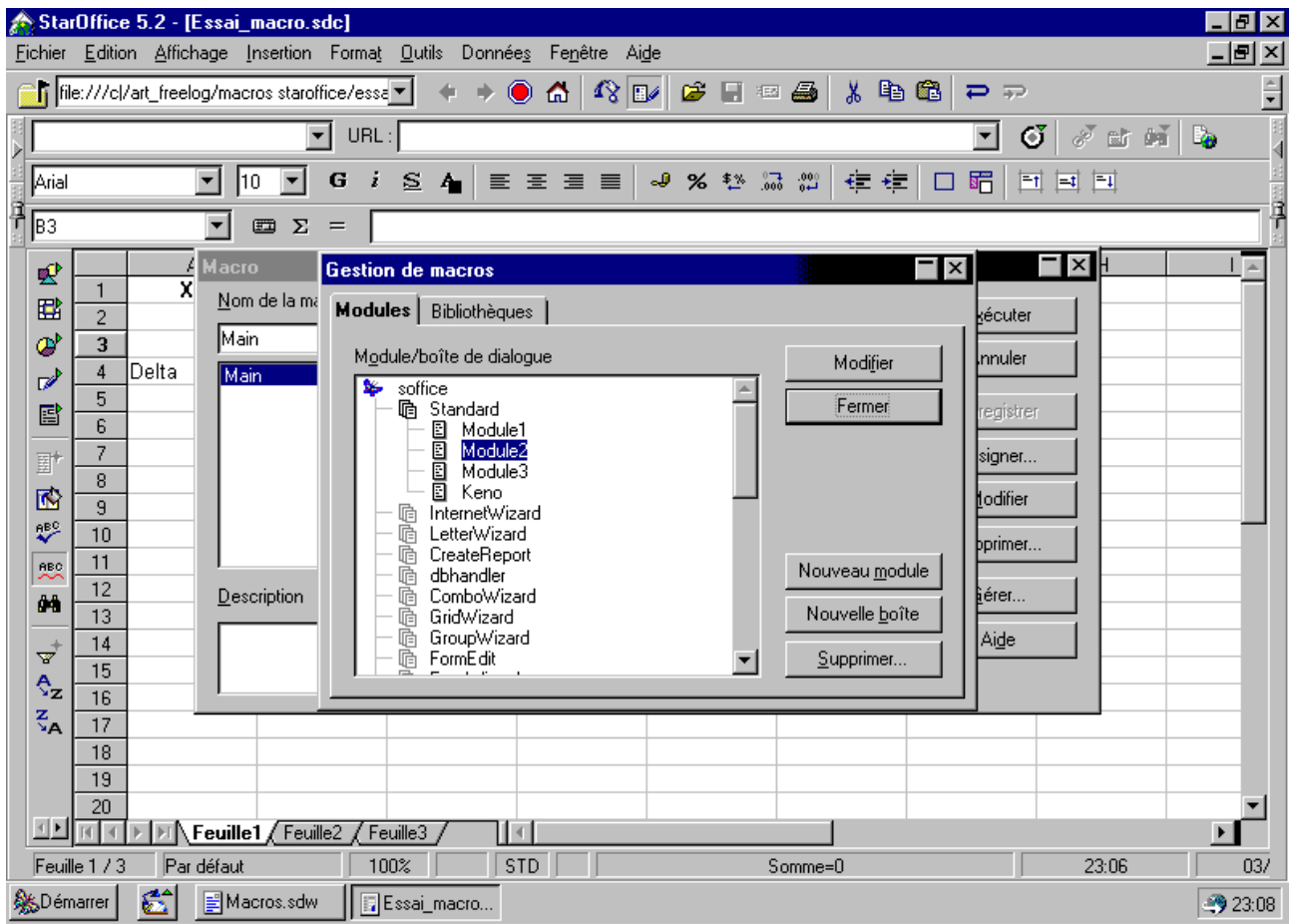
Utilisons le menu Outils>Macro (fig3)



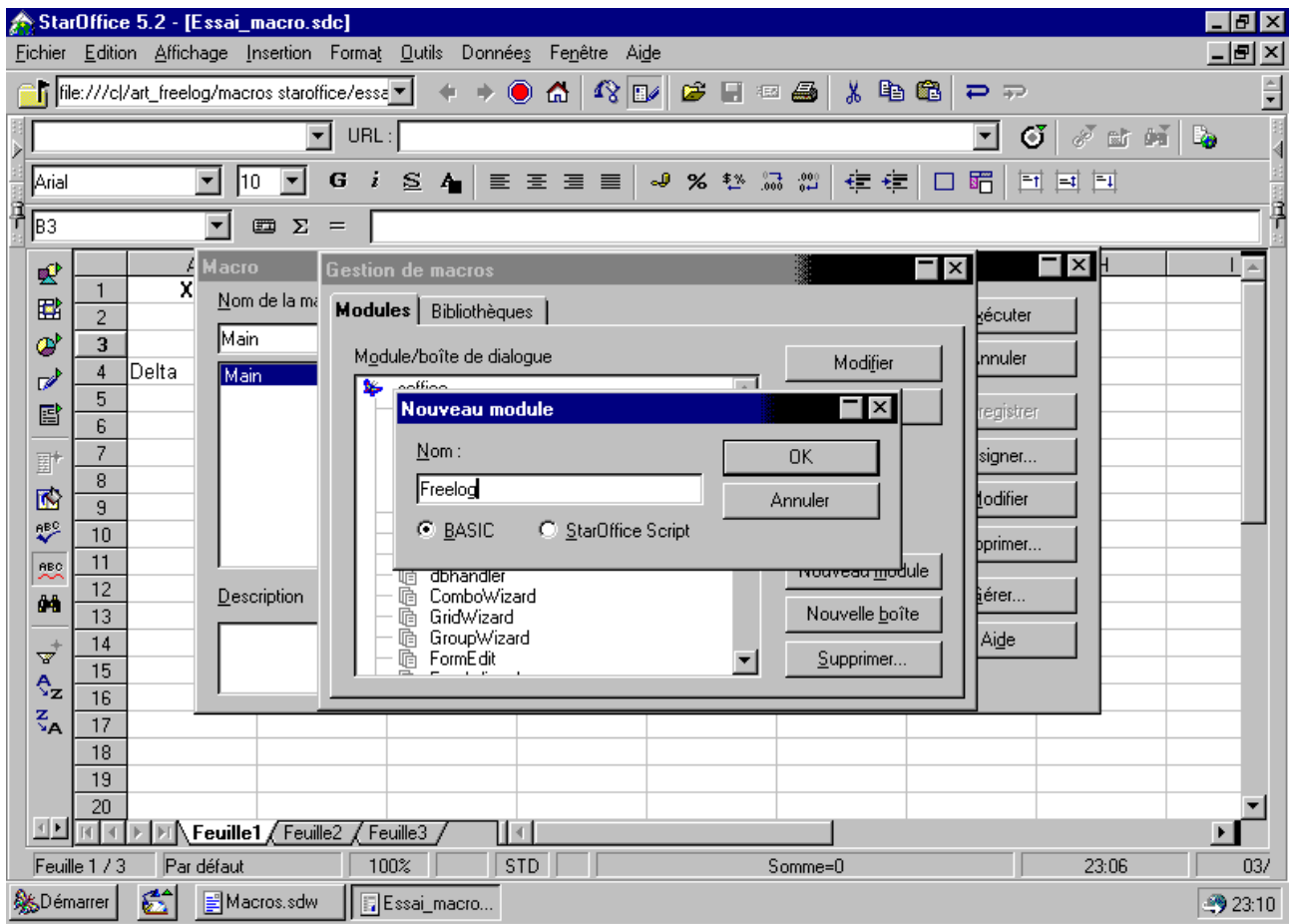
Un sous-menu apparaît alors pour que nous puissions créer notre nouveau module (fig4).



Et, bien entendu, nous choisissons l'option „gérer“ pour qu'apparaisse enfin la boîte de dialogue qui va nous demander de nommer notre macro. C'est la première étape de notre développement. Cette façon de faire devrait rappeler de bons souvenirs aux „anciens“ (fig5).

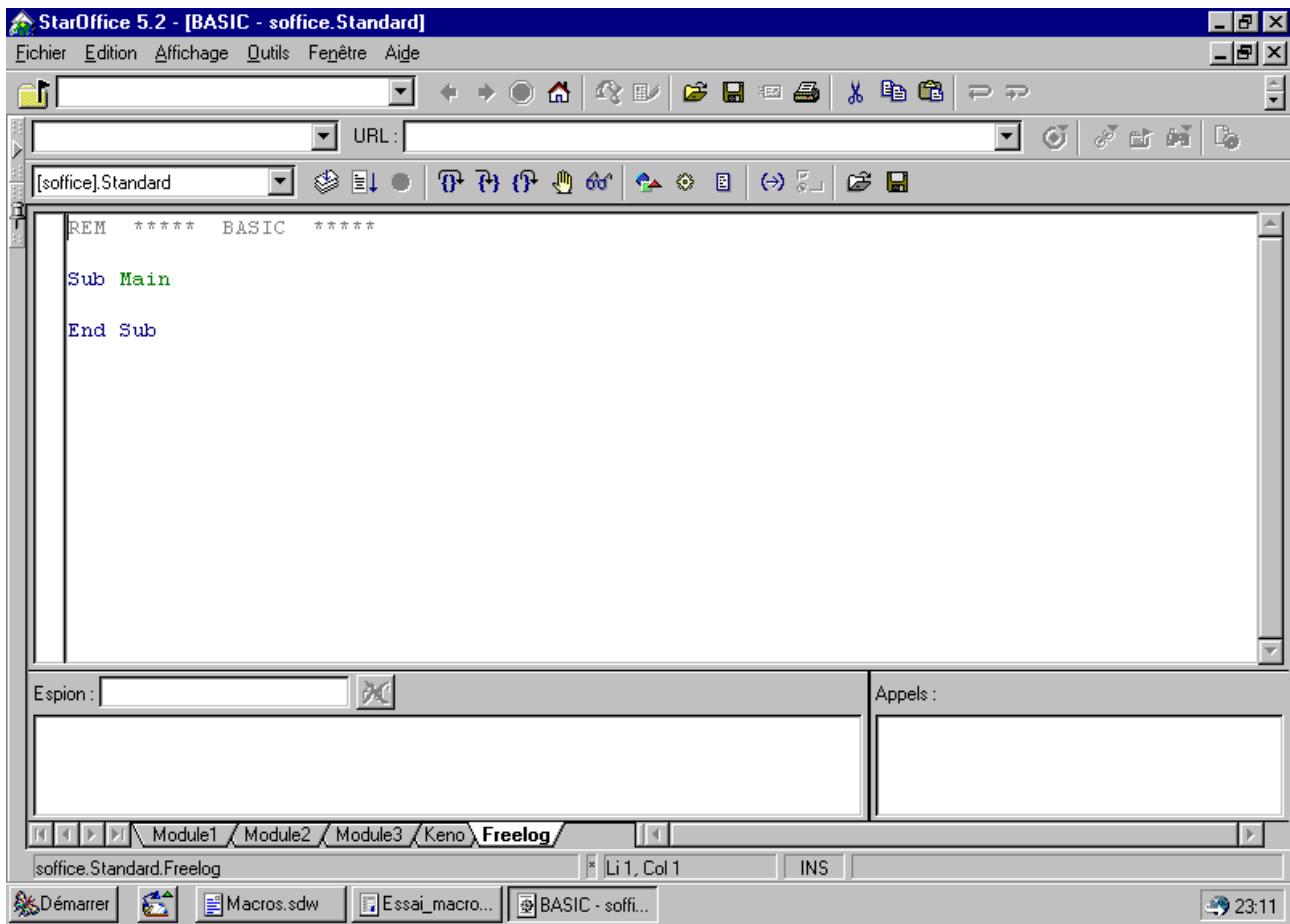


Et nous pouvons enfin, triomphalement nommer notre macro. Il est vrai que lorsque nous procédons ainsi en mode pas à pas, le cheminement à l'air fastidieux, voire complexe. Mais à l'utilisation réelle, cette manipulation ne prend que quelques secondes (fig6).



Une fois notre macro nommée, nous arrivons sur ce bon vieux basic. L'éditeur s'ouvre pour nous permettre nos saisies (fig7).





Nous allons enfin pouvoir travailler. A partir de maintenant, comme nous sommes de retour sous un éditeur classique pour le développement, nous nous concentrerons sur les instructions et non plus sur l'aspect visuel. Donc, nous ne mettrons plus de zolies copies d'écran. Concentrez-vous...

#### Tour d'horizon

Pour faciliter votre apprentissage, jusqu'ici, le mode opératoire a été disséqué. Désormais, c'est à vous de jouer. Première constatation : une fois sous l'éditeur, nous avons une multitude d'onglets. Comme vous vous en doutez, il s'agit des autres macros qui sont là disponibles pour vos yeux éblouis. Cliquez sur les onglets correspondants, et vous pourrez les lire. De plus, les instructions copier/coller fonctionnent parfaitement, ainsi que leurs raccourcis clavier. Ainsi, vous pourrez sans difficulté dupliquer des portions de code pour les réutiliser dans d'autres applications...

Si nous examinons notre fenêtre, nous voyons que l'EDI (l'interface de développement) a placé le cadre global des instructions. Il ne nous reste qu'à compléter de façon judicieuse les instructions pour créer notre première application. Au passage, notez la coloration syntaxique qui rend le procédé d'écriture de code plus convivial (pour ceux qui voudraient s'amuser, il est possible de modifier les couleurs attribuées pour chaque type de données : cherchez ;-)). Au risque de nous répéter encore et encore, utilisez les remarques (REM ou ') pour documenter votre programme et en faciliter la maintenance. Aujourd'hui, vous savez parfaitement pourquoi vous avez écrit telle ou telle portion de code de telle façon. Mais vous l'oublierez vite. Il est donc vital de laisser des clés de lectures précises aux points cruciaux de votre application, surtout en Basic, dont la souplesse le rend facilement déstructuré. C'est le revers de la médaille.

#### A vos claviers

Une précision importante : une macro qui tourne (et qui s'emballe ;-)) peut être arrêtée par la séquence : Ctrl+Alt+Q. Une autre astuce, vitale : lorsque vous êtes coincés, placez le curseur en

plein milieu de l'instruction qui vous pose un problème, puis appuyez sur la touche F1. Il ne vous reste plus qu'à prier, car l'aide en ligne de StarOffice est son point faible : armez vous de courage et de persévérance pour chercher ce dont vous avez besoin. Souvent, vous aurez besoin de connaissances de la langue de Shakespeare.

Maintenant que les gros jalons sont posés, amusons-nous avec les détails. Avec StarBasic, nous avons le choix : soit nous travaillons avec la feuille de calcul, soit nous utilisons notre macro de manière indépendante. Ne soyons pas enquinés par ce dilemme, nous allons faire les 2 en même temps... Hé oui, c'est possible... Voyons un peu le code à saisir et décortiquons-le.

Ouf, enfin du code

Enfin, on y arrive... On va enfin pouvoir utiliser les deux index de ses zolies mimines pour faire des automatisations... Notre première partie portera sur la récupération des données (valeurs à utiliser) à partir des cellules du tableau que nous avons créé. Notre première ligne sert à activer(désactiver si elle est omise) la feuille de calcul.

```
[essai_macro:1].Activate ( )
```

Entre les crochets, on reconnaît le nom de notre feuille de calcul qui est ainsi activée. A ce propos, lorsque vous lancerez la macro, il est possible que vous ayez un message d'erreur. En effet, cette instruction suppose que ce soit la feuille de calcul qui soit active au lancement de la macro. Pensez à bien vous positionner dessus le cas échéant.

Ensuite, il convient de procéder au positionnement de la cellule active. Cela revient à dire que l'on indique nommément (par ses coordonnées) la cellule qui doit être active.

```
ActiveWindow.GoToCell( "$A$2" )
```

Si vous avez de bons réflexes, vous verrez, au lancement de la macro, la cellule active se déplacer de sa position par défaut, à celle que nous venons de désigner.

Et nous pouvons ainsi procéder à la récupération de la valeur souhaitée pour faire nos calculs ultérieurs. Nous créons la variable recA, et nous lui affectons la valeur contenue dans la cellule précédemment sélectionnée.

```
recA=Selection.Value
```

Juste pour le fun, nous procédons à l'affichage de la valeur de la cellule de la façon la plus classique en Basic : Print.

```
print "La valeur qui se trouve dans la cellule A2 de la feuille de calcul est :", recA
```

La syntaxe simplissime se passe de commentaire. Raffinement, la phrase est écrite dans une box.

Cette manipulation était décrite pour l'exemple, et vous permettre de récupérer ainsi certaines valeurs. Mais il serait plus sympa de faire une saisie dans des boîtes de dialogue. Ce que nous allons faire...

Oh, oui, encore...

Petits coquins, vous voulez encore plus de code à saisir ... Alors, on y va. Cette fois, nous allons ouvrir des boîtes de saisie pour affecter nos données à 3 variables appelées A, B, C.

```
A=InputBox("Entrez la valeur de A","Saisie 1","")
```

```
B=InputBox("Entrez la valeur de B","Saisie 2","")
```

```
C=InputBox("Entrez la valeur de C","Saisie 3","")
```

Juste pour vous faire voir la meilleure convivialité du système, par rapport à une feuille de calcul classique, on affiche une boîte de confirmation de la saisie.

```
print "Vous avez saisi : A= ",A," B= ",B," C= ",C
```

Bien, nous arrivons sur la fin de notre réalisation. Il ne reste plus qu'à procéder à un transtypage.

```
AC=int(A)
```

```
BC=int(B)
```

```
CC=int(C)
```

Pourquoi ? Tout simplement parce que la valeur que vous venez de saisir, même s'il s'agit pour vous d'une variable numérique est considérée par le système comme étant un texte. Il faut donc la reconverter en nombre. A partir de là, les fantaisies les plus folles nous sont permises : on peut

calculer le fameux discriminant, delta.

```
delta = (BC*BC)-(4*(AC*CC))
```

Il ne reste plus alors qu'à l'afficher.

```
print "Delta = ",delta
```

Ouais, bon...

Certes, nous avons déployé beaucoup d'efforts, mais pour arriver quasiment au même résultat que sur une simple feuille de calcul, me direz-vous... Que nenni, vous rétorquerai-je. Car maintenant, selon la valeur du discriminant, on doit trouver différentes solutions à notre équation. Comment faire cela sur notre bonne feuille de calcul ? On peut le faire, mais c'est assez fastidieux et mal présenté. Avec notre macro, 3 lignes suffisent :

```
if delta<0 then print "Pas de racine réelle"
```

```
if delta=0 then print "Une racine double"
```

```
if delta>0 then print "Deux racines réelles distinctes"
```

Simple et convivial, c'est le principe de la macro. Les box s'afficheront pour vous préciser le résultat. Enfantin, et beaucoup plus facile que de faire la même chose directement sur la feuille de calcul. Nos variables peuvent désormais servir à tout calcul ultérieur. Pour afficher les racines finales, si nous utilisons la feuille de calcul, il nous faudrait tâtonner, en vérifiant à deux fois les coordonnées des cellules. Ici, il suffira d'écrire  $racine1=(-BC-sqr(delta))/(2*AC)$  et d'utiliser un print pour afficher le résultat (d'ailleurs, on vous laisse le faire, il faut que vous travaillez vous aussi;-) ). Pas de prise de tête, gain de temps, convivialité, voilà l'avantage. D'ailleurs, on est (trop) gentils, on vous a mis sur la macro du CD, le calcul des deux racines pour vous montrer les possibilités de codage de la structure if/then sur une seule ligne et if/then/endif dans le cas où il y ait plusieurs lignes de codes dans la structure conditionnelle.

Cerise sur le gâteau

Décidément, avec FREELOG, on vous en donne toujours plus. Puisque nous avons commencé avec une feuille de calcul, nous finirons sur une feuille de calcul. Nous avons vu au début de cette réalisation, comment récupérer les données dans les cellules. Il est donc juste qu'à la fin, nous voyons comment écrire notre résultat directement dans la feuille de calcul. Ainsi, la boucle est bouclée. Comme précédemment, on se positionne sur la cellule voulue par un `ActiveWindow.GoToCell`, et on affiche par `Selection.Insert`.

```
ActiveWindow.GoToCell( "$B$4" )
```

```
Selection.Insert( "delta = " )
```

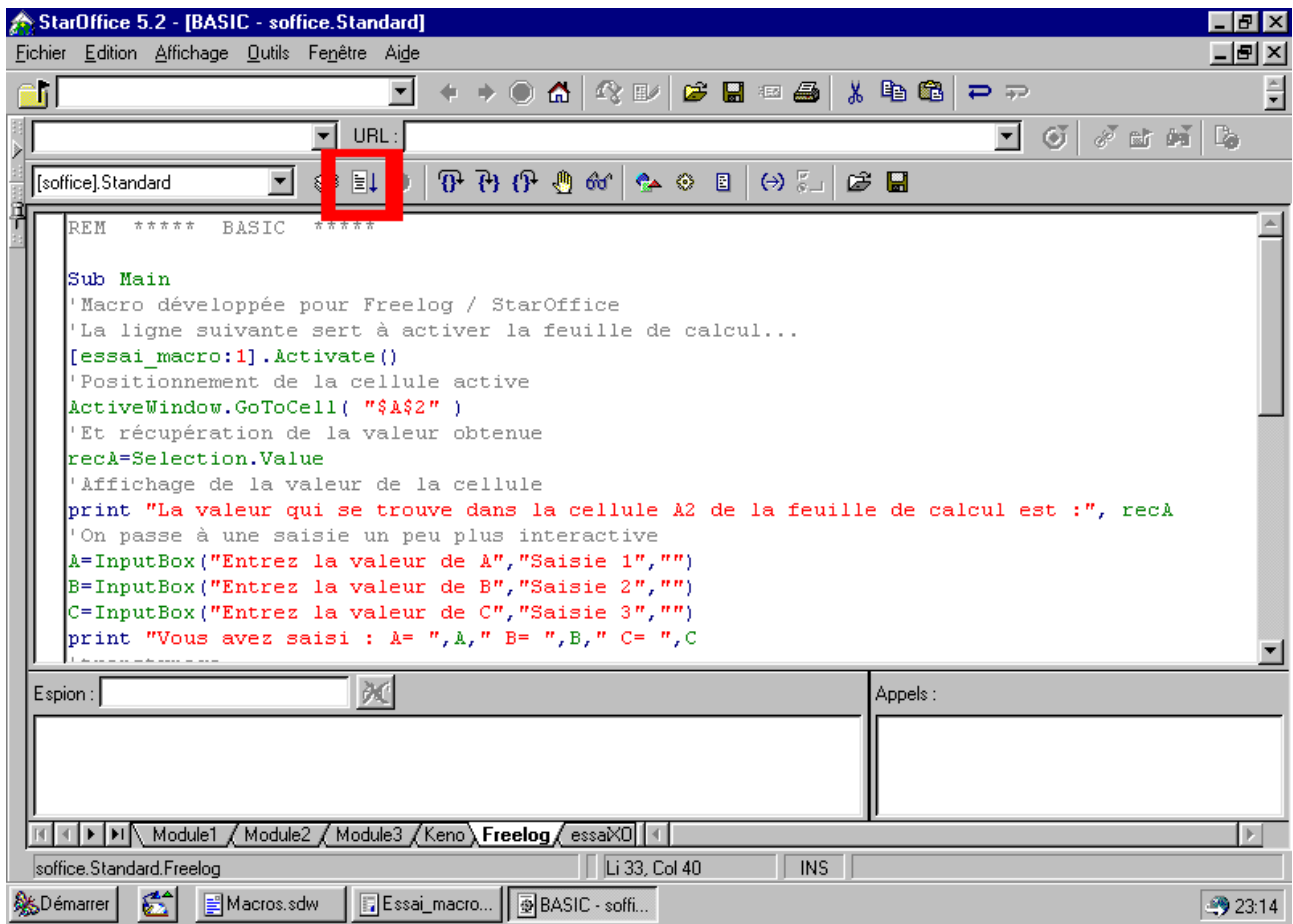
```
ActiveWindow.GoToCell( "$B$5" )
```

```
Selection.Insert( delta )
```

Remarquez que dans le premier cas, les guillemets permettent d'afficher le mot delta, alors que dans le second cas, sans les guillemets, nous affichons la valeur de la variable delta.

Cerise sur la cerise du gâteau

Et si on vous indiquait incidemment comment lancer cette fameuse macro ? Il suffit de cliquer sur le bon bouton (Fig8).



## Conclusion

Nous n'avons vu qu'un tout petit aperçu des possibilités de StarBasic, mais nous espérons qu'il vous incitera à vous lancer dans l'aventure de la programmation. Notre réalisation de ce jour n'était qu'un prétexte pour vous faire découvrir le potentiel de la programmation des macros. Pour ceux qui veulent simplement automatiser une tâche précise, il existe une possibilité d'enregistrement d'actions faites à la souris. Consultez l'aide en ligne. Pour ceux qui veulent aller plus loin, nous vous avons montré comment on peut travailler à partir de la feuille de calcul, aussi bien que travailler uniquement avec la macro. Nous ferons ultérieurement d'autres réalisations avec StarBasic (notamment avec la structure for/next) pour vous montrer que les macros c'est très simple, pourvu qu'on procède avec méthode. Encore une fois, notre CD n'attend que vos réalisations : faites-vous des crampes aux doigts.