

Ecran de démarrage



TI-Nspire



Cette touche permet d'ouvrir une nouvelle page de calcul.

Exemple de mise en œuvre :


On utilise considère la suite définie par :
$$\begin{cases} U_0 = \frac{1}{3} \\ U_{n+1} = 10 \times U_n - 3 \end{cases}$$

Normalement cette suite est constante.

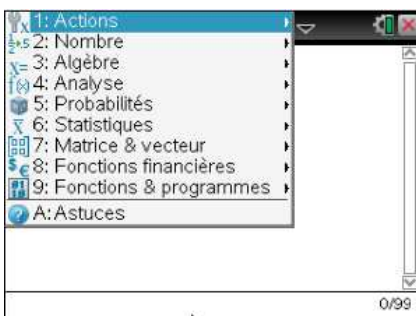
On désire réaliser un programme qui demande la valeur du premier terme U_0 et le nombre n termes à afficher de la suite.

1°) Accès au mode programme :

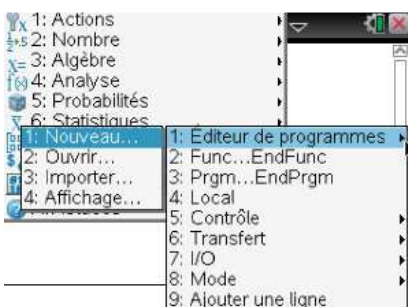


On a une page vierge. On appuie sur la touche 

On a accès ainsi à 9 menus.

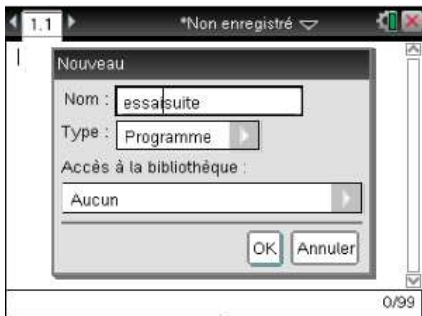



On choisit alors le menu
9 : Fonctions et programmes

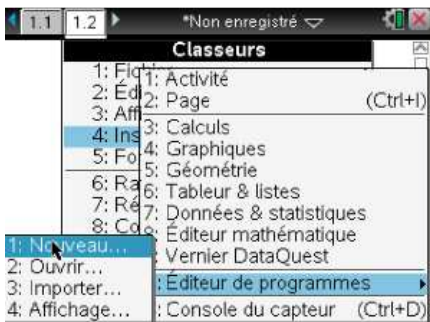


On déroule alors un menu dans lequel on choisit
1 : Editeur de programmes puis
1 : Nouveau.

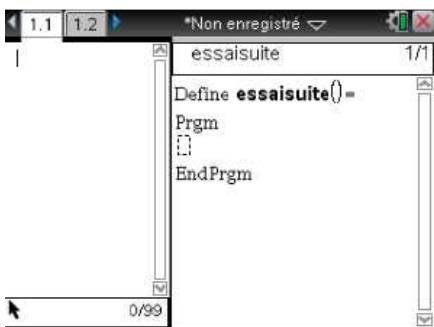
Par ce menu, on peut rappeler un ancien programme ...



Il faut alors choisir un nom pour le programme puis valider avec 



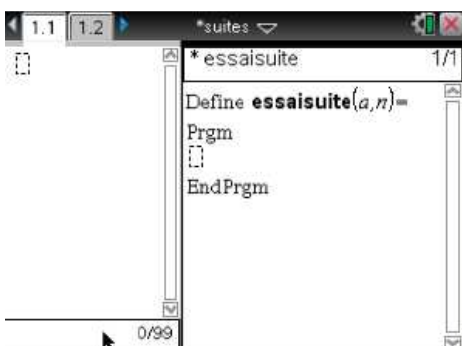
Remarque : on peut aussi accéder à un programme à partir du menu outils accessible en cliquant sur la flèche située à côté de « Non enregistré » dans notre cas puis en choisissant dans le menu choisir 4 : Insertion puis A : Editeur de programmes puis 1 : Nouveau. On arrive alors sur la même procédure que précédemment.



On a alors ouvert l'éditeur de programme dans un partage d'écran. On peut agrandir la fenêtre de programmation en réduisant ma case de droite. La calculatrice définit automatiquement le début et la fin du programme.

2°) Généralités :

Pour tout programme, il faut indiquer le nom des paramètres utilisés lors de l'appel de ce programme en les deux parenthèses à côté du nom du programme. S'il y en a plusieurs, il faut les séparer avec une virgule. Ici il y en a deux : le premier terme de la suite et le nombre de termes désirés. Si aucun paramètre n'est appelé, on laisse les parenthèses vides.

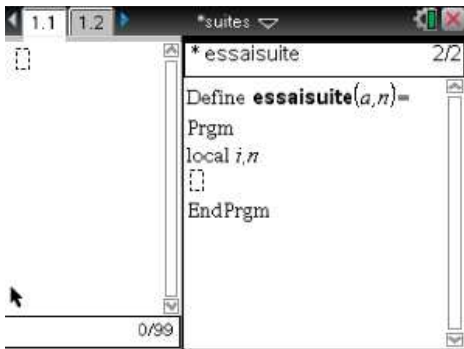


On commence par renseigner le nom des paramètres utilisés et appelés par le programme.
Le premier terme de la suite U_0 que l'on a nommé a
Le nombre de terme à calculer que l'on a nommé n

L'étoile * qui apparait à droite du nom du programme précise que ce dernier a été modifié mais pas sauvegardé !

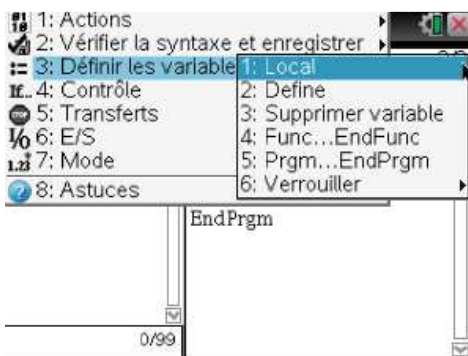
Lorsque l'on construit un programme, on doit décider des variables que l'on va utiliser et les indiquer au programme. Cela se fait au moyen de l'instruction *Local*.

Le programme pourrait fonctionner sans cette instruction *Local*, mais dans ce cas toutes les opérations faites sur les mêmes variables affecteraient les variables de même nom pouvant exister par ailleurs dans cette activité, à l'extérieur de ce programme. On a alors des variables globales.



Dans notre programme, on peut utiliser deux variables locales :
 Une première pour le compteur des termes de la suite. On l'appelle *i*.
 Un second pour afficher le terme de la suite. On l'appelle *u*.
 Pour cela, on tape : local i,n dans la case vide en dessous de Prgm.

Puis on valide avec

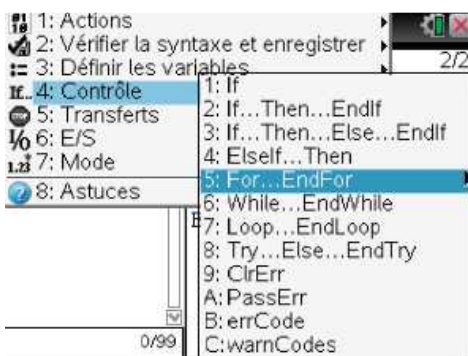


On peut aussi accéder aux instructions à l'aide de la touche

- 1 : Actions (nouveau, ouvrir, importer ...)
- 2 : Vérifier la syntaxe et enregistrer
- 3 : Définir les variables (Local, ce qui nous intéresse)
- 4 : Contrôle (boucles...)
- 5 : Transferts (renvoie...)
- 6 : E/S (pour écrire ...)
- 7 : Mode (choisir le format des nombres, angles ...)

3°) Programme :

On étudie plus particulièrement notre suite. Elle comprend une boucle et doit faire afficher les termes de la suite.



On peut utiliser une boucle de type

Pour ... fin pour.

On peut procéder de deux façons. Soit tout écrire, soit utilisé la touche puis le point 4 : Contrôle et dans le menu déroulé le point 5 : For ... EndFor

L'instruction est de la forme :

For *compteur,début,fin,pas*

EndFor

```

*essaisuite 3/4
Define essaisuite(a,n)=
Prgm
Local i,n
For i,1,n,1
[]
EndFor
EndPrgm
    
```

Dans notre exemple, i varie de 1 à n avec un pas de 1 donc, on a :
 For $i,1,n,1$

 EndFor.

Pour faire fonctionner la suite, il fallait affecter à u le premier terme choisi (ici a). On peut insérer une ligne dans le programme à n'importe quel moment, il suffit de se positionner sur la ligne précédente et d'appuyer sur **enter**. Ce que l'on fait pour donner à u la valeur a . On tape $u := a$

```

*essaisuite 5/5
Define essaisuite(a,n)=
Prgm
Local i,n
u:=a
For i,1,n,1
[]
EndFor
EndPrgm
    
```

Il ne reste plus qu'à renseigner la boucle avec le calcul des termes
 $u := 3 \times u - 10$
 Et faire afficher les termes
 L'instruction **Disp** est alors utilisée. On place entre des guillemets le texte à écrire et on peut intercaler à l'aide de virgules les valeurs

```

essaisuite 4/6
Define essaisuite(a,n)=
Prgm
Local i,n
u:=a
For i,1,n,1
u:=10·u-3
Disp "le",i,"ième terme vaut :",u
EndFor
EndPrgm
    
```

Le programme est alors terminer. Pour pouvoir le lancer, il faut **impérativement** faire **vérifier la syntaxe et l'enregistrer**.
 On accède à cette fonction grâce à la touche **menu** puis choisir le 2 :
 On valide grâce à **enter**

```

1: Actions
2: Vérifier la syntaxe et enregistrer (Ctrl+B)
3: Vérifier la syntaxe
4: Contrôle
5: Transferts
6: E/S
7: Mode
8: Astuces
    
```

Si aucun message n'apparaît c'est que tout va bien. La calculatrice relève les erreurs de syntaxe ... automatiquement. On va donc pouvoir lancer le programme. Si l'étape précédente n'a pas été faite, cela est impossible !

Résultats :

On constate que tout va bien dans la barre située au dessus du programme qui signale l'enregistrement.

Pour lancer le programme, on va dans la fenêtre de gauche. Si l'on veut les 5 premiers termes de la suite, l'on tape `essaisuite(1/3,5)` puis

```

"essaisuite" enregistrement effectué
Define essaisuite(a,n)=
Prgm
Local i,n
u:=a
For i,1,n,1
u:=10·u-3
Disp "le ",i,"ième terme vaut :"
EndFor
EndPrgm
    
```

```

essaisuite(1/3,5)
"essaisuite" enregistrement effectué
Define essaisuite(a,n)=
Prgm
Local i,n
u:=a
For i,1,n,1
u:=10·u-3
Disp "le ",i,"ième terme
EndFor
EndPrgm
    
```

On constate que la suite est constante et vaut $\frac{1}{3}$.

```

le 4ième terme
vaut : 1/3
le 5ième terme
vaut : 1/3
Terminé
    
```

Remarque : on peut recommencer et prendre une valeur approchée de $\frac{1}{3}$ à savoir 0.333 ou 0.3333 etc. ...

```

le 1ième terme
vaut : 0.33
le 2ième terme
vaut : 0.3
le 3ième terme
vaut : 0.
le 4ième terme
vaut : -3.
    
```