Dans cette première leçon de l'Unité 4 vous allez étudier le concept de boucle et la structure et l'utilisation de la boucle **For(…)**.

Objectifs:

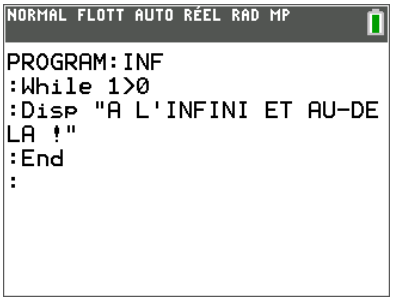
- Comprendre l'utilisation des boucles.
- Utiliser la boucle **For(…)** pour générer une liste de valeurs.

Indication : Il y a en TI-Basic trois types fondamentaux de boucles : **For**, **While** et **Repeat**. Une structure de boucle donne à un programme la capacité de traiter un ensemble d'instructions plusieurs fois de suite, soit par itération sur une séquence de valeurs (comme dans la boucle **For**), soit jusqu'à ce qu'une condition spécifique soit remplie (ou non) comme dans **While** et **Repeat**. Dans cette Unité chaque leçon ne traite qu'une de ces structures. Il est aussi possible d'utiliser les archaïques instructions **Lbl** et **Goto** pour faire une boucle, mais ceci conduit à de mauvaises habitudes et des erreurs de programme peuvent survenir si ce n'est pas fait proprement, donc nous éviterons toutes références à ces instructions. Cependant, l'instruction **Lbl** est utilisée et, en fait, requise en conjonction avec l'instruction **Menu** pour concevoir des menus personnalisés à l'intérieur d'un programme. Pour les instructions conditionnelles et les boucles, l'instruction **Goto** n'est pas nécessaire du tout.

Les programmes peuvent se compliquer quand il devient nécessaire de faire un mélange de toutes les structures de contrôle (instructions **If** et boucles) dans un programme pour satisfaire à des algorithmes plus complexes. C'est ce qui rend la programmation FUN !

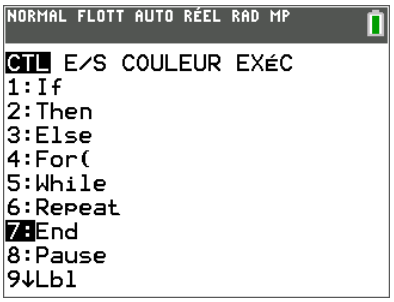
Boucles

Une boucle est une méthode permettant de répéter un ensemble d'instructions. Tout langage de programmation possède au moins une structure de boucle. La structure de boucle possède un moyen pour revenir en arrière dans un programme en un point précédent. Le TI-Basic a trois différents types de boucles. Une boucle infinie ne se termine jamais.



Une boucle *infinie*. Pourquoi ?

Pour arrêter un programme en cours d'exécution, appuyez sur on. Vous verrez s'afficher les options 'Quitter' et 'Voir'. 'Quitter' vous permet de retourner à l'écran de calcul, alors que 'Voir' vous permet de retourner dans l'Éditeur de programme à l'endroit où le programme a été interrompu.



Les trois types de boucles du TI-Basic sont :

For( ) … **End**
While <condition vraie> … **End**
Repeat <*jusqu'à* condition vraie> … **End**

Le reste de cette leçon ne traite que de la boucle **For**( ).

La boucle For(…)

Structure : **For**(*variable*, *valeur de départ*, *valeur de fin*)
corps de la boucle
End

Exemple : **For**(A,1,10)
Disp A ← *corps de la boucle*
End

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COMPT10
:For(A,1,10)
:Disp A
:End
```

Note :
*L'instruction **For**( ) a besoin d'une variable (la variable de contrôle de la boucle), d'une valeur de départ et d'une valeur de fin, séparées par des virgules. Les valeurs de départ et de fin peuvent être des variables. Le corps de la boucle peut comporter autant d'instructions que l'on veut mais ne doit pas modifier la variable de contrôle. Les boucles sont effectuées de la valeur de départ à la valeur de fin avec un incrément standard de 1.*

```
NORMAL FLOTT AUTO RÉEL RAD MP
                              3
                              4
                              5
                              6
                              7
                              8
                              9
                             10
                           Fait
```

Indication : Les boucles se terminent lorsque la variable *dépasse* la valeur de fin. Si vous affichez A (**Disp A**) après la boucle, vous verrez que sa valeur est 11, et non 10.

Boucle For avec incrément autre que 1

Il y a un quatrième argument optionnel pour l'instruction **For( )** : l'*incrément*. L'incrément est la valeur ajouté à chaque passage de boucle à la variable de contrôle. Par défaut la valeur est 1.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:COMPT10
:For(A,1,10,3)
:Disp A
:End
```

Indication : Quand l'incrément est négatif, la valeur de départ doit être supérieure à la valeur de fin, et la boucle s'arrête quand la variable est *inférieure* à la valeur de fin.

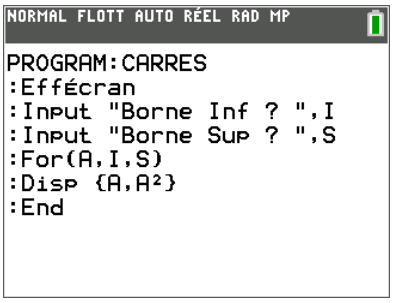
For(A,1,10,3) débute avec **A**=1, puis ajoute 3 à **A** à chaque passage de boucle. La boucle s'arrête quand **A** est plus grand que 10. L'incrément peut être un nombre négatif.

For(B,10,0,-1) fait un compte à rebours de 10 à 0.

```
NORMAL FLOTT AUTO RÉEL RAD MP
prgmCOMPT10
                              1
                              4
                              7
                             10
                           Fait
```

Programmation avec For(…)

Écrivons un programme qui affiche un tableau de nombres et de leur carré.
L'utilisateur peut entrer les bornes inférieure et supérieure de l'ensemble des nombres.
La partie difficile est l'affiche des paires de nombres sur la même ligne ! On peut faire cela grâce aux listes.



Note :
I *et* **S** *sont utilisés pour représenter la borne inférieure et la borne supérieure.*
L'instruction **For( )** *utilise les valeurs de* **I** *et de* **S**.
Les accolades des listes s'obtiennent en appuyant sur 2nde (*et sur* 2nde).

Indication : Utiliser une liste constitue une méthode commode pour afficher plus d'une valeur sur une même ligne de l'écran de calcul.

Exécutez le programme, entrez la borne inférieure puis la borne supérieure pour le tableau.

Si les listes passent trop vite, vous pouvez ajouter une instruction **Pause** après l'instruction **Disp** et avant le **End**.

Challenge :
Utilisez une structure **If … Then… End** pour faire une **Pause** toutes les 5 paires de nombres. Rappelez-vous la technique utilisée dans la précédente Unité.

Exemple de réponse :

```
:If (A-I+1)/5=partEnt((A-I+1)/5)
:Pause "Appuyez sur entrer"
```