



# **Guide de programmation TI-Basic pour la calculatrice graphique TI-84 Plus CE**

Pour obtenir la version la plus récente de cette documentation, consultez [education.ti.com/guides](http://education.ti.com/guides).

## Informations importantes

Sauf disposition contraire stipulée dans la licence qui accompagne un programme, Texas Instruments n'émet aucune garantie expresse ou implicite, y compris sans s'y limiter, toute garantie implicite de valeur marchande et d'adéquation à un usage particulier, concernant les programmes ou la documentation, ceux-ci étant fournis "tels quels" sans autre recours. En aucun cas, Texas Instruments ne peut être tenue responsable vis à vis de quiconque pour quelque dommage de nature spéciale, collatérale, fortuite ou indirecte occasionné à un tiers, en rapport avec ou découlant de l'achat ou de l'utilisation desdits matériels, la seule et exclusive responsabilité de Texas Instruments, pour quelque forme d'action que ce soit, ne pouvant excéder le montant indiqué dans la licence du programme. Par ailleurs, la responsabilité de Texas Instruments ne saurait être engagée pour quelque réclamation que ce soit en rapport avec l'utilisation desdits matériels par toute autre tierce partie.

© 2006 - 2016 Texas Instruments Incorporated

## Sommaire

<b>Informations importantes</b> .....	<b>ii</b>
<b>Nouveautés</b> .....	<b>1</b>
Nouveautés du Guide de programmation TI-Basic pour la calculatrice graphique TI-84 Plus CE v5.2 .....	1
<b>Introduction au TI-Basic sur votre TI-84 Plus CE</b> .....	<b>2</b>
Qu'est-ce qu'un programme ? .....	2
<b>Premières activités :</b> .....	<b>3</b>
Programmer la formule donnant le volume d'un cylindre à partir de son rayon et de sa hauteur .....	3
Création d'un NOUVEAU programme .....	3
Nommer le programme .....	4
Entrer des commandes .....	5
Afficher le Volume calculé .....	6
Exécution d'un programme .....	6
Trouver le volume .....	7
<b>Créer et supprimer des programmes</b> .....	<b>8</b>
Versions du système d'exploitation et Programmation .....	8
Création d'un nouveau programme .....	8
Gestion de la mémoire et suppression d'un programme .....	9
Augmenter la mémoire disponible .....	9
<b>Entrer des lignes de commande et exécuter des programmes</b> .....	<b>12</b>
Entrée d'une ligne de commande d'un programme .....	12
Exécution d'un programme .....	12
Interruption d'un programme .....	13
<b>Modification de programmes</b> .....	<b>14</b>
Modification d'un programme .....	14
<b>Copier et renommer les programmes</b> .....	<b>15</b>
Copier et renommer un programme .....	15
Faire défiler les menus PGRM EXEC et PRGM EDIT .....	15
<b>Instruction PGRM CTL (Control)</b> .....	<b>16</b>
Menu PRGM CTL .....	16

<b>Instructions PRGM E/S (Entrée/Sortie)</b> .....	<b>30</b>
Menu PRGM E/S .....	30
<b>Instructions PRGM COULEUR</b> .....	<b>42</b>
Menu PRGM COULEUR .....	42
<b>Instructions PRGM EXÉC</b> .....	<b>43</b>
Appel d'autres programmes comme sous-programmes .....	43
Appel d'un programme à partir d'un autre programme .....	43
<b>Instructions PRGM HUB</b> .....	<b>45</b>
Instructions du Menu TI-Innovator™ HUB .....	45
Menu TI-Innovator™ HUB .....	45
<b>Exécution d'un programme en assembleur</b> .....	<b>61</b>
<b>Informations générales</b> .....	<b>63</b>
Support et service de Texas Instruments .....	63
Informations Garantie et Assistance .....	63
<b>Index</b> .....	<b>64</b>

# Nouveautés

## *Nouveautés du Guide de programmation TI-Basic pour la calculatrice graphique TI-84 Plus CE v5.2*

### Modifications du Menu du guide de programmation :

Tous les éléments de cette liste sont des entrées nouvelles ou mises à jour dans le Guide de programmation TI-Basic pour la calculatrice graphique TI-84 Plus CE v5.2.

#### Menu PRGM CNTRL

- [Pause](#)
- [Wait](#)

#### Menu PRGM E/S

- [eval\(](#)
- [expr\(](#)
- [Get\(](#)
- [Send\(](#)
- [toString\(](#)
- [String>Equ\(](#)

#### Menu PRGM Hub

- [Send\("Set...](#)
- [Send\("READ...](#)
- [Settings...](#)
- [Wait](#)
- [Get\(](#)
- [eval\(](#)
- [Send\("CONNECT-OUTPUT...](#)
- [Send\("CONNECT-INPUT...](#)
- [Ports...](#)
- [Send\("RANGE...](#)
- [Send\("AVERAGE...](#)
- [Send\("DISCONNECT-OUTPUT...](#)
- [Send\("DISCONNECT-INPUT...](#)
- [Manage...](#)

## Introduction au TI-Basic sur votre TI-84 Plus CE

Vous pouvez utiliser TI-Basic pour créer un programme sur votre calculatrice graphique. Vous pouvez créer un programme qui contrôlera une sortie souhaitée ou contrôler une expérience, comme un jeu.

### ***Qu'est-ce qu'un programme ?***

Un programme est un ensemble d'une ou plusieurs lignes de commande, contenant chacune une ou plusieurs instructions. Lorsque vous exécutez un programme, la TI 84 Plus CE exécute chaque instruction sur chaque ligne de commande dans l'ordre dans lequel vous les avez entrées. Le nombre et la taille des programmes que la TI 84 Plus CE peut stocker sont limités uniquement par la mémoire disponible.

Pour créer un programme, entrez simplement des lignes de commande en utilisant l'éditeur de programme. Le programme sera exécuté depuis l'écran de calcul. Utilisez ce guide pour apprendre comment créer, modifier et supprimer des programmes.

**Astuce :** Utilisez l'aide du Catalogue en appuyant sur [ + ] sur la plupart des commandes pour vous aider à remplir les arguments corrects pour les commandes avant de les copier dans l'éditeur de programme.

Tandis que vous progressez dans la programmation, un éditeur de programme TI-Basic est également disponible dans le logiciel TI Connect™ CE. Vous pouvez utiliser l'espace de travail de l'éditeur de programme dans TI Connect™ CE pour créer des programmes, pour envoyer des programmes vers une calculatrice connectée par USB, pour tester vos programmes et pour sauvegarder des programmes dans votre ordinateur. L'espace de travail de l'éditeur de programme dans TI Connect™ CE permet de copier, couper, coller et annuler des commandes.

**Remarque :** L'éditeur de programme dans la calculatrice ne contient pas de fonctionnalités d'édition comme copier, couper, coller ou annuler. Sur la calculatrice, n'oubliez pas que vous ne pouvez pas annuler un [clear] ou un [del].

## Premières activités :

### **Programmer la formule donnant le volume d'un cylindre à partir de son rayon et de sa hauteur**

A partir du rayon et de la hauteur d'un cylindre, vous pouvez calculer son volume à l'aide de cette formule. Cette activité vous permet d'écrire un programme pour demander les valeurs du rayon et de la hauteur d'un cylindre, afin de vous permettre ensuite d'en calculer le volume.

La formule pour le volume d'un cylindre est

$$= \pi R^2 H \text{ unités cubiques}$$

Où

V = Volume

R = Rayon de la base

H = Hauteur du cylindre

Ce programme peut être utile pour diverses activités comme :

- Fournir un tableau avec de nombreuses valeurs de Rayon et de Hauteur et demander aux élèves de remplir la colonne du Volume
- Exécuter un programme pour remplir les valeurs de volume dans le tableau

Quelques questions à examiner :

- (si les élèves ne connaissent pas la formule), quel modèle voyez-vous dans les valeurs de Volume pour leur faire deviner la formule ?
- Quel est le plus grand Volume trouvé ?
- De combien le Volume augmente-t-il si la Hauteur augmente d'une unité ?
- De combien le Volume augmente-t-il si le Rayon augmente d'une unité ?

Exécuter un programme plusieurs fois comme outil permet une analyse rapide de problèmes plus complexes.

### **Création d'un NOUVEAU programme**

1. Appuyez sur **PRGM** **▶** **▶** pour afficher le menu **PRGM NEW**.

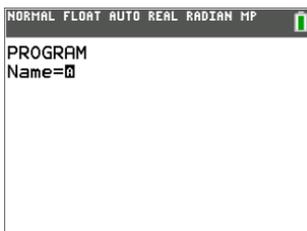
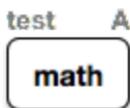


## Nommer le programme

1. Appuyez sur **[ENTER]** pour sélectionner **1:Create New**.

L'invite **Name=** est affichée et [2nd] [A-lock] (alpha-lock) est activé.

**Astuce:** Les caractères alphabétiques se trouvent en haut à droite des touches du clavier et sont copiés lorsque vous appuyez sur [alpha] or [2nd] [A-lock] avant d'appuyer sur la touche principales.



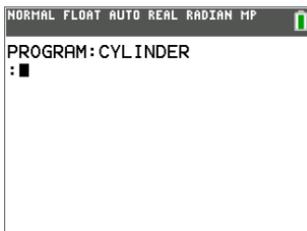
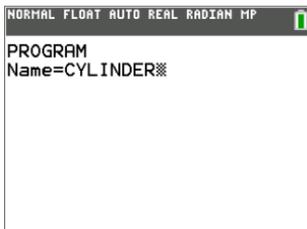
2. Appuyez sur **CYLINDER**, puis appuyez sur **[ENTER]** pour nommer le programme **CYLINDER**.

**Astuce:** Les noms de programme peuvent être composés de huit caractères maximum. Le premier caractère doit être une lettre. Veuillez remarquer le curseur du damier sur l'écran lorsque le maximum est atteint.

3. Appuyez sur **[ENTER]** et vous êtes maintenant dans l'éditeur de programme.

Les deux points ( : ) dans la première colonne de la deuxième ligne indique le début d'une ligne de commande.

**Remarque :** Sur la calculatrice, les lignes de commande ne sont pas numérotées comme lors de l'utilisation de l'éditeur de Programme TI Connect™.

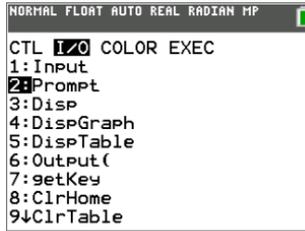


## Entrer des commandes

La personne utilisant votre programme devra entrer les valeurs de Rayon et de Hauteur. Vous utiliserez la commande **Prompt**.

1. Appuyez sur **PRGM** **▶** pour accéder au menu E/S (Entrée/Sortie).
2. Appuyez sur **◀** pour mettre la commande **Prompt** en surbrillance.

**Remarque** : Pour cette exemple, vous utiliserez l'aide du Catalogue pour illustrer l'aide à la syntaxe des arguments intégrée dans la calculatrice. Si vous connaissez déjà les arguments d'une commande, vous pouvez choisir un élément et le copier dans l'éditeur de Programme sans utiliser l'aide du Catalogue.



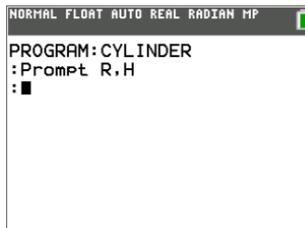
```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL 1/0 COLOR EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
8:CirHome
9↓CirTable
```

3. Le numéro d'élément de menu de **Prompt** est mis en surbrillance, appuyez sur **+**. Utilisez l'aide du Catalogue pour l'éditeur de syntaxe (si nécessaire) La syntaxe pour les arguments de **Prompt** est affichée sous la ligne d'édition tandis que les variables sont séparées par des virgules. Tout argument entre crochets [ ] est optionnel, **Prompt** a donc besoin d'au moins un nom de variable.



```
NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
Prompt R,H
variableA,variableB
....,variable n
PASTE| ESC
```

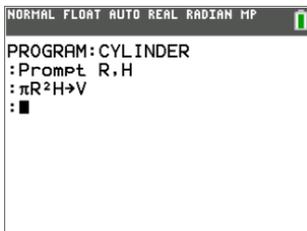
4. Appuyez sur **alpha** **R** **.** **alpha** **H** pour entrer les noms de variables pour le Rayon et la Hauteur.
5. Appuyez sur **[PASTE]** (**trace**) pour copier la commande avec les arguments dans l'éditeur de Programme. Appuyez sur **[ESC]** (**graph**) pour retourner au dernier emplacement du curseur sans copier.
6. De retour dans l'éditeur de Programme, appuyez sur **enter** pour déplacer le curseur vers la ligne de commande suivante.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CYLINDER
:Prompt R,H
:█
```

Stockez la formule du volume d'un cylindre.

7. Pour entrer l'expression  $[\pi] R^2 H$  et stocker la valeur dans la variable V, appuyez sur  $[2nd] [\pi] [\alpha] R [x^2] [\alpha] H [sto\rightarrow] [\alpha] V [enter]$ .



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CYLINDER
:Prompt R,H
:πR²H→V
:

```

## Afficher le Volume calculé

Créez une ligne de commande pour afficher le volume calculé :

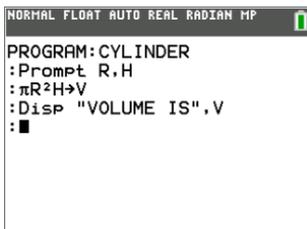
1. Appuyez sur  $[prgm]$   $\blacktriangleright$  **3** pour sélectionner 3:Disp depuis le menu **PRGM E/S**.

**Disp** est copié sur la ligne de commande.

**Astuce** : Souvenez-vous que vous pouvez appuyer sur  $[+]$  sur la plupart des commandes pour utiliser l'aide du Catalogue pour l'éditeur de syntaxe pour voir les arguments corrects pour les commandes.

2. Appuyez sur  $[2nd] [A-lock] ["]$  **VOLUME EST**  $["] [\alpha] [, ] [\alpha] V [enter]$

Ceci affichera le texte **VOLUME EST** sur une ligne et la valeur calculée de V sur la ligne suivante de l'écran de calcul lorsque vous exécutez le programme.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CYLINDER
:Prompt R,H
:πR²H→V
:DISP "VOLUME IS",V
:

```

## Exécution d'un programme

Votre programme est terminé ! Vous pouvez maintenant exécuter le programme à partir de l'écran de calcul

1. Appuyez sur  $[2nd] [quit]$  pour afficher l'écran de calcul.
2. Appuyez sur  $[prgm]$  pour afficher le menu **PRGM EXEC**.

Les éléments de ce menu sont les noms des programmes stockés.



```
NORMAL FLOAT AUTO REAL RADIAN MP
EXEC EDIT NEW
1:CYLINDER

```

- Appuyez sur `enter` pour copier **prgm** **CYLINDER** à l'emplacement actuel du curseur. (Si **CYLINDER** n'est pas l'élément 1 de votre menu **PRGM EXEC**, déplacez le curseur vers **CYLINDER** avant d'appuyer sur `enter`.)

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmCYLINDER
```

## Trouver le volume

Pour trouver le volume du cylindre de 1,5 cm de rayon et de 3 cm de hauteur, suivez les étapes suivantes.

- Appuyez sur `enter` pour (exécuter) le programme.
- Entrez **1,5** pour **R** et appuyez sur `enter`
- Entrez **3** pour **H** et appuyez sur `enter`.

Le texte **VOLUME EST**, la valeur de **V** et **Fait** s'affichent.

Le volume du cylindre s'affiche à 8 décimales près  
21. 20575041 cm cube.

- À ce stade, pour relancer le programme, appuyez sur `enter` et répétez pour les différentes valeurs de **R** et de **H**.

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmCYLINDER
R=?1.5
H=?3
VOLUME IS          21.20575041
.....
Done
```

# Créer et supprimer des programmes

Cette section décrit comment créer des programmes et comment supprimer des programmes.

## Versions du système d'exploitation et Programmation

- Les programmes créés avec la TI-84 Plus OS 2.55MP et plus anciens ou la TI-83 Plus 1.19 OS et plus anciens s'exécuteront sur la TI-84 Plus CE ; cependant, il peut y avoir des affichages inattendus sur le TI-84 Plus CE résultant de la haute résolution de l'écran. Vous devez tester vos programmes existants sur la TI-84 Plus CE et ajuster en conséquence les arguments des commandes. En particulier, les arguments de toutes les commandes qui apparaissent sur un graphique doivent être ajustés aux emplacements des pixels désirés sur la zone du graphique. Les programmes s'affichant dans l'écran de calcul doivent s'exécuter comme prévu.
- L'exécution des programmes peut s'effectuer en mode classique ou en mode MathPrint™.
- Les menus de raccourcis sont disponibles partout où le menu MATH est accessible.
- Les modèles MathPrint™ ne sont pas disponibles pour les programmes. Toutes les saisies et l'affichage des résultats s'effectuent en mode classique.
- Vous pouvez utiliser les fractions dans les programmes, mais il est recommandé de tester le programme afin de s'assurer que les résultats obtenus correspondent à vos attentes.
- L'espacement de l'affichage peut être légèrement différent en mode MathPrint™ par rapport au mode classique. Si vous préférez l'espacement du mode classique, réglez le mode à l'aide d'une commande dans votre programme. Les captures d'écran des exemples présentés dans ce chapitre ont été réalisées en mode MathPrint™.
- L'aide à la syntaxe est intégrée sur la TI-84 Plus CE. En mode édition de programme, appuyez sur

**Remarque :** Appuyez sur  $\oplus$  lorsqu'une commande est sélectionnée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

## Création d'un nouveau programme

Pour créer un nouveau programme, suivez ces étapes.

1. Appuyez sur  $\boxed{\text{PRGM}}$   $\boxed{1}$  pour afficher le menu **PRGM NOUVEAU**.



- Appuyez sur **[ENTER]** pour sélectionner **1:Create New**. L'invite **Name=** est affichée et l'alpha lock est activé.
- Appuyez sur une lettre de A à Z ou  $\theta$  pour entrer le premier caractère du nouveau nom de programme.

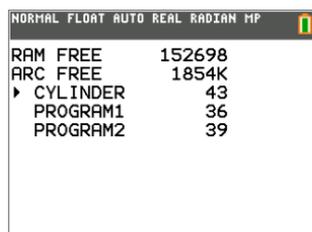
**Remarque** : le nom du programme peut comporter de 1 à 8 caractères. Le premier caractère doit être une lettre de A à Z ou  $\theta$ . Les caractères suivants peuvent être des lettres, des chiffres ou  $\theta$ .

- Entrez de 0 à 7 lettres, chiffres ou  $\theta$  pour compléter le nom du nouveau programme.
- Appuyez sur **[ENTER]**. L'éditeur de programme est affiché.
- Entrez une ou plusieurs commandes de programme.
- Appuyez sur **[2nd] [QUIT]** pour quitter l'éditeur de programme et retourner à l'écran de calcul.

## ***Gestion de la mémoire et suppression d'un programme***

Pour vérifier si la mémoire disponible est suffisante pour un programme que vous souhaitez entrer :

- Appuyez sur **[2nd] [MEM]** pour afficher le menu **MEMORY (MÉMOIRE)**.
- Choisissez **2:Mem Management/Delete** pour afficher le menu **MEMORY MANAGEMENT/DELETE**.
- Choisissez **7:Prgm** pour afficher l'éditeur **PRGM**.



NORMAL FLOAT AUTO REAL RADIAN MP	
RAM FREE	152698
ARC FREE	1854K
▶ CYLINDER	43
▶ PROGRAM1	36
▶ PROGRAM2	39

La TI-84 Plus CE indique la quantité de mémoire en octets.

## ***Augmenter la mémoire disponible***

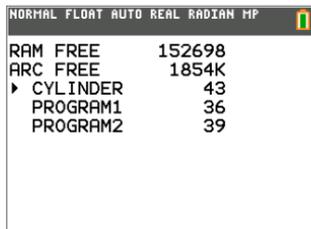
Vous pouvez augmenter la mémoire disponible de deux façons. Vous pouvez supprimer un ou plusieurs programme ou vous pouvez archiver certains programmes.

**Pour augmenter la mémoire disponible en supprimant un programme spécifique :**

- Appuyez sur **[2nd] [MEM]** puis sélectionnez **2:Mem Management/Delete** depuis le menu **MEMORY**.



2. Choisissez **7:Prgm** pour afficher les fichiers de type programme.



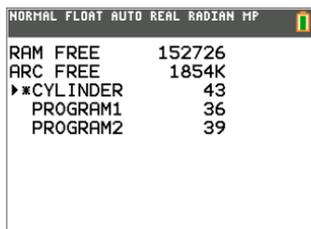
3. Appuyez sur  $\uparrow$  et  $\downarrow$  [ALPHA] pour déplacer le curseur de sélection ( $\blacktriangleright$ ) près du programme que vous voulez supprimer et appuyez sur [DEL]. Le programme est supprimé de la mémoire

**Remarque :** Vous recevrez un message vous demandant de confirmer cette action de suppression. Choisissez **2:yes** pour continuer.

Pour quitter l'écran d'édition **PRGM** sans supprimer de programme, appuyez sur  $\square$  [QUIT], qui affiche l'écran de calcul.

#### Pour augmenter la mémoire disponible en archivant un programme :

1. Appuyez sur  $\square$  [MEM] puis sélectionnez **2:Mem Management/Delete** depuis le menu **MEMORY**.
2. Choisissez **2:Mem Management/Delete** pour afficher le menu **MEMORY MANAGEMENT/DELETE** .
3. Choisissez **7:Prgm...** pour afficher les fichiers de type programme.



4. Appuyez sur [ENTER] pour archiver le programme. Une astérisque apparaîtra à gauche du programme pour indiquer que ce programme est archivé.

Pour désarchiver un programme dans cet écran, placez le curseur près du programme archivé et appuyez sur **ENTER**. L'astérisque disparaît.

**Remarque** : Il vous est impossible de modifier ou d'exécuter les programmes archivés. Pour modifier ou exécuter un programme archivé, vous devez d'abord le désarchiver.

# Entrer des lignes de commande et exécuter des programmes

Cette section décrit comment entrer des lignes de commande et exécuter des programmes.

## ***Entrée d'une ligne de commande d'un programme***

Sur une ligne de commande, vous pouvez entrer toute commande, instruction ou expression que vous pouvez exécuter depuis l'écran de calcul. Dans l'éditeur de programme, chaque nouvelle ligne de commande commence avec deux-points. Pour entrer plus d'une instruction ou expression sur une ligne de commande, séparez-les avec deux-points.

**Remarque :** Une ligne de commande peut être plus longue que la largeur de l'écran.

Dans l'éditeur de programme, vous pouvez afficher et sélectionner depuis les menus. Vous pouvez retourner à l'éditeur de programme depuis un menu de deux façons.

- Sélectionnez une option de façon à insérer l'élément dans la ligne de commande courante.  
— ou —
- Appuyez sur  u.

Pour terminer une ligne de commande, appuyez sur **ENTER**. Le curseur se positionne sur la ligne de commande suivante.

Les programmes peuvent accéder aux variables, aux listes, aux matrices et aux chaînes de caractères sauvegardées dans la mémoire. Si un programme stocke une nouvelle valeur dans une variable, une liste, une matrice ou une chaîne de caractères, le programme change la valeur dans la mémoire pendant l'exécution.

Vous pouvez appeler un autre programme comme sous-programme.

## ***Exécution d'un programme***

Pour exécuter un programme, commencez sur une ligne vierge de l'écran de calcul et suivez ces étapes.

1. Appuyez sur **PRGM** pour afficher le menu **PRGM EXEC**.
2. Sélectionnez un nom de programme dans le menu **PRGM EXEC**. *prgmname* est collé dans l'écran de calcul (par exemple, **prgmCYLINDER**).
3. Appuyez sur **ENTER** pour exécuter le programme. Pendant que le programme s'exécute, l'indicateur d'occupation est activé.

Last Answer (**Ans**) est mis à jour pendant l'exécution du programme. Last Entry n'est pas mis à jour pendant que chaque commande est exécutée.

La TI-84 Plus CE cherche les erreurs pendant l'exécution d'un programme. Elle ne cherche pas les erreurs pendant que vous entrez un programme.

## ***Interruption d'un programme***

Pour annuler l'exécution d'un programme, appuyez sur **ON**. Le menu **ERR:BREAK** s'affiche.

- Pour revenir dans l'écran de calcul, sélectionnez **1:Quitter**.
- Pour aller à l'endroit où l'interruption s'est produite, sélectionnez **2:Goto**.

## Modification de programmes

Dans cette section, vous suivrez les étapes suivantes pour modifier un programme. Cette section décrit comment insérer et supprimer une ligne de commande.

### Modification d'un programme

Pour modifier un programme stocké, suivez ces étapes.

1. Appuyez sur **PRGM**  pour afficher le menu **PRGM EDIT**.
2. Sélectionnez un nom de programme dans le menu **PRGM EDIT**. Les neuf premières lignes du programme sont affichées.

**Remarque** : L'éditeur de programme n'affiche pas de ↓ pour indiquer qu'un programme continue en dehors de l'écran.

3. Modifiez les lignes de commande du programme.
  - Déplacez le curseur à l'emplacement approprié, puis supprimez, écrasez ou insérez.
  - Appuyez sur **CLEAR** pour effacer toutes les commandes de programme sur la ligne de commande (les deux points d'entrée restent), puis entrez une nouvelle commande de programme.

**Remarque** : Pour déplacer le curseur au début d'une ligne de commande, appuyez sur **2nd** ; pour le déplacer à la fin, appuyez sur **2nd** . Pour faire défiler le curseur vers le bas de sept lignes de commande, appuyez sur **ALPHA** . Pour faire défiler le curseur vers le haut de sept lignes de commande, appuyez sur **ALPHA** .

### Insérer et Supprimer des Lignes de Commande

Pour insérer une nouvelle ligne de commande n'importe où dans le programme, placez le curseur à l'endroit où vous voulez insérer la nouvelle ligne, appuyez sur **2nd** **[INS]**, puis appuyez sur **ENTER**. Deux points indiquent une nouvelle ligne.

Pour supprimer une ligne de commande, placez le curseur sur la ligne, appuyez sur **CLEAR** pour effacer toutes les instructions sur la ligne, puis appuyez sur **DEL** pour supprimer la ligne de commande, y compris les deux points.

## Copier et renommer les programmes

Cette section décrit comment copier et renommer un programme et comment parcourir les menus.

### **Copier et renommer un programme**

Pour copier toutes les lignes de commande d'un programme dans un nouveau programme, suivez les étapes 1 à 5 pour Créer un Nouveau programme, puis suivez ces étapes.

1. Appuyez sur **[2nd] [RCL]**. **RCL** s'affiche sur la dernière ligne de l'éditeur de programme dans le nouveau programme.
2. Appuyez sur **[PRGM]** pour afficher le menu **PRGM EXEC**.
3. Sélectionnez un nom dans le menu. **prgmnom** est collé sur la dernière ligne de l'éditeur de programme.
4. Appuyez sur **[ENTER]**. Toutes les lignes de commande du programme choisi sont copiées dans le nouveau programme.

Copier les programmes a au moins deux applications pratiques.

- Vous pouvez créer un modèle pour les groupes d'instructions que vous utilisez fréquemment.
- Vous pouvez renommer un programme en copiant son contenu dans un nouveau programme.

**Remarque** : Vous pouvez également copier toutes les lignes de commande d'un programme existant dans un autre programme en utilisant **RCL**.

### **Faire défiler les menus PRGM EXEC et PRGM EDIT**

La TI-84 Plus CE trie les éléments de menu **PRGM EXEC** et **PRGM EDIT** automatiquement par ordre alphanumérique. Chaque menu marque uniquement les 10 premiers éléments à l'aide des touches 1 à 9, puis 0.

Pour passer au premier nom de programme commençant par un caractère particulier alpha ou  $\theta$ , appuyez sur **[ALPHA]** [lettre de A à Z ou  $\theta$ ].

**Remarque** : Depuis le haut du menu **PRGM EXEC** ou **PRGM EDIT**, appuyez sur **[▲]** pour vous déplacer à la fin. À partir du bas du menu, appuyez sur **[▼]** pour en afficher le début. Pour faire défiler le curseur vers le haut dans les sept éléments du menu, appuyez sur **[ALPHA] [▼]**. Pour faire défiler le curseur vers le bas dans les sept éléments du menu, appuyez sur **[ALPHA] [▲]**.

# Instruction PGRM CTL (Control)

Cette section les instructions **PGRM CTL** (Control).

## Menu PGRM CTL

Pour afficher le menu **PGRM CTL**, appuyez sur **[PRGM]** seulement depuis l'éditeur de programme.

**Astuce importante** : Pour retrouver rapidement une commande, utilisez **[alpha]** **[↑]** ou **[alpha]** **[↓]** pour vous déplacer entre les écrans.

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7:End
8:Pause
9↓Lbl
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
9↑Lbl
0:Goto
A:Wait
B:IS>(
C:DS<(
D:Menu(
E:PRgm
F:Return
G↓Stop
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
D↑Menu(
E:PRgm
F:Return
G:Stop
H:DelVar
I:GraphStyle(
J:GraphColor(
K:OpenLib(
L:ExecLib
```

**CTRL**    E/S    COULEUR EXEC    HUB

### Description

- |    |        |   |
|----|--------|---|
| 1: | If     | Crée un test conditionnel :                         |
| 2: | Then   | Exécute les commandes lorsque <b>If</b> est vraie.  |
| 3: | Else   | Exécute les commandes lorsque <b>If</b> est fausse. |
| 4: | For(   | Crée une boucle incrémentée.                        |
| 5: | While  | Crée une boucle conditionnelle.                     |
| 6: | Repeat | Crée une boucle conditionnelle.                     |

7: End	Signifie la fin d'un bloc :
8: Pause	Met en pause l'exécution d'un programme.
9: Lbl	Définit un label.
0: Goto	Va vers un label.
A: Wait	Suspend l'exécution d'un programme pour un temps donné.
B: IS>(	Incrémente et saute si plus grand que.
C: DS<(	Décrémente et saute si plus petit que.
D: Menu(	Définit les éléments de menu et les branches.
E: prgm	Exécute un programme comme sous-programme.
F: Return	Revient depuis une sous-programme.
G: Stop	Arrête l'exécution.
H: DelVar	Efface le contenu d'une variable depuis un programme.
I: GraphStyle(	Désigne le style de graphique à tracer.
J: CouleurGraph(	Désigne la couleur du graphique à tracer.
K: OpenLib(	Extension de TI-Basic (non disponible)
L: ExeLib(	Extension de TI-Basic (non disponible)

**Remarque :** Appuyez sur  lorsqu'une commande est surlignée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

Ces éléments de menu dirigent le flux d'un programme en cours d'exécution. Ils facilitent la répétition ou le saut d'un groupe de commande pendant l'exécution du programme. Lorsque vous sélectionnez un élément depuis le menu, le nom est collé à l'emplacement du curseur sur la ligne de commande dans le programme.

Pour revenir à l'éditeur de programmes sans sélectionner la rubrique, appuyez sur .

### Contrôle du flux d'un programme

Les instructions de contrôle d'un programme indiquent à la TI-84 Plus CE quelle prochaine commande exécuter dans un programme. **If**, **While**, et **Repeat** vérifient une condition définie pour déterminer quelle commande exécuter ensuite. Les conditions utilisent fréquemment des tests relationnels ou booléen, comme dans :

**If A<7:A+1>A**

or

**If N=1 et M=1:Goto Z**

## If

Utilisez **If** pour des tests et des branchements. Si *condition* est fausse (0), alors la *commande* suivant immédiatement **If** est sautée. Si *condition* est vraie (non nulle), alors la *commande* suivante est exécutée. Les instructions **If** peuvent être imbriquées.

```
:if condition  
:commande (si vraie)  
:commande
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP  
PROGRAM: COUNT  
:0→A  
:Lb1 Z  
:A+1→A  
:Disp "A IS ",A  
:If A≥2  
:Stop  
:Goto Z  
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP  
Pr9mCOUNT  
A IS 1  
A IS 2  
..... Done.
```

---

## If-Then

**Then** suivant un **If** exécute un groupe de *commandes* si *condition* est vraie (non nulle). **End** indique la fin du groupe de *commandes*.

```
:if condition  
:Then  
:commande (si vraie)  
:commande (si vraie)  
:End  
:commande
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP  
PROGRAM: TEST  
:1→X:10→Y  
:If X<10  
:Then  
:2X+3→X  
:2Y-3→Y  
:End  
:Disp X,Y  
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP  
Pr9mTEST  
..... Done.  
5  
17
```

## If-Then-Else

**Else** suivant **If-Then** exécute un groupe de *commandes* si *condition* est fausse (nulle).  
**End** indique la fin du groupe de *commandes*.

**:if** *condition*

**:Then**

*:commande* (si vraie)

*:commande* (si vraie)

**:Else**

*:commande* (si vraie)

*:commande* (si vraie)

**:End**

*:commande*

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TESTELSE
: Input "X=", X
: If X<0
: Then
: X^2→Y
: Else
: X→Y
: End
: Disp {X, Y}
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mTESTELSE
X=5
{5 5}
..... Done.
Pr9mTESTELSE
X=-5
{-5 25}
..... Done.
```

**Remarque** : Appuyez sur **ENTER** pour répéter le programme

---

## For(

**For(** effectue des boucles et incrémente Il incrémente la *variable* de *début* à *fin* par *incrément*. *incrément* est facultatif (la valeur par défaut est 1) et peut être négatif ( $fin < début$ ). *fin* est une valeur maximum ou minimum à ne pas dépasser **End** indique la fin de la boucle. Les boucles **For(** peuvent être imbriquées.

```
:For(variable,début,fin[,incrément])  
:commande (alors que fin non dépassée)  
:commande (alors que fin non dépassée)  
:End  
:commande
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SQUARE
:For(A,0.8,2)
:Disp A^2
:End
:█
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmSQUARE
0
4
16
36
64
.....Done
```

## While

**While** exécute un groupe de *commandes* tandis que *condition* est vraie. *condition* est souvent un test relationnel. *condition* est testée lorsque **While** est rencontré. Si *condition* est vraie (non nulle), le programme exécute un groupe de *commandes*. **End** indique la fin du groupe. Lorsque *condition* est fausse (nulle), le programme exécute chaque *commande* suivant **End**. Les instructions **While** peuvent être imbriquées.

**:While condition**

**:commande** (alors que *condition* est vraie)

**:commande** (alors que *condition* est vraie)

**:End**

**:commande**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: LOOP
:0→I
:0→J
:While I<6
:J+1→J
:I+1→I
:End
:Disp "J=",J
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Prgm LOOP
J=
.....6
.....Done
█
```

---

## Repeat

**While** répète un groupe de *commandes* tandis que *condition* est vraie (non nulle). Elle est similaire à **While**, mais *condition* est testée lorsque **End** est rencontré; le groupe de *commandes* est donc exécuté au moins une fois. Les instructions **Repeat** peuvent être imbriquées.

**:Repeat condition**

**:commande** (jusqu'à ce que *condition* soit vraie)

**:commande** (jusqu'à ce que *condition* soit vraie)

**:End**

**:commande**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: RLOOP
:0→I
:0→J
:Repeat I≥6
:J+1→J
:I+1→I
:End
:Disp "J=",J
:█
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Prgm RLOOP
J=
.....6
.....Done
█
```

---

## End

**End** indique la fin d'un groupe de *commandes*. Vous devez inclure une instruction **End** à la fin de chaque boucle **For**, **While**, ou **Repeat**. Vous devez également inclure une instruction **End** à la fin de chaque groupe **If-Then** et de chaque groupe **If-Then-Else**.

---

## Pause

**Pause** suspend l'exécution du programme, afin que vous puissiez voir les réponses ou les graphiques. Pendant la pause, l'indicateur de pause est situé dans le coin supérieur droit.

- **Pause** sans argument interrompt temporairement le programme. Si l'instruction **DispGraph** ou **Disp** a été exécutée, l'écran approprié apparaît. Appuyez sur **[ENTER]** pour reprendre l'exécution.
- **Pause** avec *valeur* affiche *valeur* sur l'écran de calcul courant. Vous pouvez faire défiler *valeur*. **Pause** [*valeur*] Appuyez sur **[ENTER]** pour reprendre l'exécution.
- **Pause** avec *valeur* et *temps* affiche la valeur sur l'écran de calcul actuel et l'exécution du programme continue pour la période spécifiée. Pour le temps seulement, utilisez **Pause " ",time** où la valeur est une chaîne de caractères vide. Temps en secondes. **Pause** *valeur,temps*.

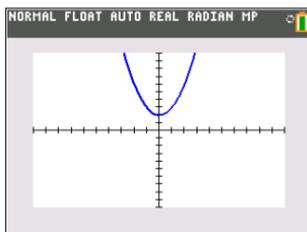
**Remarque** : Lorsque vous utilisez l'Éditeur de programme de TI Connect CE, **Pause** doit avoir un espace après la commande même si aucun argument n'est entré.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: PAUSE
:10→X
:"X^2+2"→Y1
:Disp "X=",X
:Pause
:DispGraph
:Pause
:Disp
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mPAUSE
X= 10
```



```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mPAUSE
X= 10
..... Done
```

---

## Lbl, Goto

-----

### Lbl

**Lbl** (étiquette) et **Goto** (aller à) sont utilisés ensemble pour le branchement.

**Lbl** spécifie le *label* pour une commande. *label* peut avoir un ou deux caractères (A à Z, 0 à 99, ou $\theta$ ).

### Lbl étiquette

-----

### Goto

**Goto** branche le programme à *label* lorsque **Goto** est rencontré.

### Gotoétiquette

#### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CUBE
:Lbl 99
:Input A
:If A≥100
:Stop
:Disp A³
:Pause
:Goto 99
:
```

#### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrmCUBE
?2
?3
?105
..... Done
```

---

## Wait

**Wait** suspend l'exécution d'un programme pour un temps donné. Le temps maximum est de 100 secondes. Pendant le temps de pause, l'indicateur d'occupation est situé dans le coin supérieur droit.

### Wait temps

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:WAIT
:ClrDraw
:AxesOff:F0ff
:TextColor(MAGENTA)
:Text(2,2,"HELLO WORLD"
:Wait 5
:TextColor(GREEN)
:Text(24,2,"BYE!"
:
```

### Output : "Bye!" s'affiche après 5 secondes.

```
NORMAL FLOAT AUTO REAL RADIAN MP
HELLO WORLD
BYE!
```

---

### IS>{

**IS>{** (incrémente et saute) ajoute 1 à *variable*. Si la réponse est  $> \text{valeur}$  (qui peut être une expression), la *commande* suivante est sautée; si la réponse est  $\leq \text{valeur}$ , la *commande* suivante est exécutée. *variable* ne peut pas être une variable système

**:IS>{**(*variable,valeur*)

*:commande* (si la réponse  $\leq \text{valeur}$ )

*:commande* (si la réponse  $> \text{valeur}$ )

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:ISKIP
:7→A
:IS>(A,6)
:DISP "NOT > 6"
:DISP "> 6"
:█
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmISKIP
> 6
..... Done
█
```

**Remarque : IS>{** n'est pas une instruction de boucle.

---

### DS<{

**DS<{** (décrémente et saute) soustrait 1 de *variable*. Si la réponse est  $> \text{valeur}$  (qui peut être une expression), la *commande* suivante est sautée; si la réponse est  $\leq \text{valeur}$ , la *commande* suivante est exécutée. *variable* ne peut pas être une variable système

**:DS>{**(*variable,valeur*)

*:commande* (si la réponse  $\leq \text{valeur}$ )

*:commande* (si la réponse  $> \text{valeur}$ )

### Programme

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: DSKIP
:1>A
:DS<(A,6)
:DISP "> 6"
:DISP "NOT> 6"
:█
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgrmDSKIP
NOT> 6
..... Done.
█
```

**Remarque :** DS<( n'est pas une instruction de boucle.

## Menu(

**Menu(** définit un branchement dans un programme. Si **Menu(** est rencontré pendant l'exécution d'un programme, l'écran du menu est affiché avec les éléments de menu spécifiés, l'indicateur de pause est activé et l'exécution est interrompue jusqu'à ce qu'un élément de menu soit sélectionné.

Le *titre* du menu est entre guillemets ( " ). Jusqu'à neuf paires d'éléments de menu sont autorisées. Chaque paire est constituée d'un élément *texte* (également entre guillemets) à afficher comme menu de sélection et d'un élément *label* auquel brancher si vous sélectionnez la sélection de menu correspondante.

**Menu("titre","texte1",label1,"texte2",label2,...)**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TOSSDICE
:Menu("TOSS DICE","FAIR DI
CE",A,"WEIGHTED",B)
:Lbl A
:Lbl B
:█
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
TOSS DICE
1:FAIR DICE
2:WEIGHTED
```

Le programme ci-dessus s'interrompt jusqu'à ce que **1** ou **2** soit sélectionné. Si vous sélectionnez **2**, par exemple, le menu disparaît et le programme continue l'exécution à **Lbl B**.

## prgm

Utilisez **prgm** pour exécuter d'autres programmes comme sous-programmes. Lorsque vous sélectionnez **prgm**, il est collé à l'emplacement du curseur. Entrez les caractères pour écrire le *nom* d'un programme. Utiliser **prgm** est équivalent à sélectionner des

programmes existants depuis le menu **PRGM EXEC**; cependant, il vous permet également d'entrer le nom d'un programme que vous n'avez pas encore créé.

**prgm***nom*

**Remarque** : Vous ne pouvez pas entrer directement le nom du sous-programme lorsque vous utilisez **RCL**. Vous devez copier le nom depuis le menu **PRGM EXEC**.

---

## Return

**Return** sort d'un sous-programme et l'exécution retourne au programme appelant, même s'il est rencontré avec des boucles imbriquées. Toutes les boucles sont terminées. Un **Return** existe à la fin de tout programme appelé comme sous-programme. Dans le programme principal, **Return** arrête l'exécution et retourne à l'écran de calcul.

---

## Stop

**Stop** arrête l'exécution d'un programme et retourne à l'écran de calcul. **Stop** est facultatif à la fin d'un programme.

---

## DelVar

**DelVar** supprime de la mémoire le contenu de *variable*.

**EffVar** *variable*



## GraphStyle(

**GraphStyle(** désigne le style du graphique à tracer. *fonction#* est le numéro du nom de la fonction Y= dans le mode graphique courant. *graphstyle* est un chiffre de 1 à 7 correspondant au style du graphique, comme indiqué ci-dessous

1 =  $\backslash$  (Thin)

5 =  $\dagger$  (Path)

2 =  (Thick)  
3 =  (Shade above)  
4 =  (Shade below)

6 =  (Animate)  
7 =  (Dot-Thick)  
8 =  (Dot-Thin)

### GraphStyle(*function#*,*graphstyle*)

Par exemple, **GraphStyle(1,5)** en mode **Func** définit le style du graphique pour Y1 à  $\psi$  (chemin; 5).

Tous les styles ne sont pas disponibles dans les modes graphiques.

---

### CouleurGraph

**GraphColor** désigne la couleur du graphique à tracer. *function#* est le numéro du nom de la fonction Y= dans le mode graphique courant. *color#* est un chiffre de 10 à 24 correspondant à la couleur du graphique, comme indiqué ci-dessous :

Numéro de couleur	Nom de couleur
10	BLEU
11	ROUGE
12	NOIR
13	MAGENTA
14	VERT
15	ORANGE
16	MARRON
17	BLEU MRN
18	BLEU CLR
19	JAUNE
20	BLANC
21	GRIS CLR
22	GRIS MOY
23	GRIS
24	GRIS FON

Vous pouvez également choisir une couleur dans le menu **VAR[S]** (sous-menu **couleur**).

NORMAL FLOAT AUTO REAL RADIAN MP					NORMAL FLOAT AUTO REAL RADIAN MP				
CTL	I/O	COLOR	EXEC	HUB	CTL	I/O	COLOR	EXEC	HUB
1:		BLUE			7↑		BROWN		
2:		RED			8:		NAVY		
3:		BLACK			9:		LTBLUE		
4:		MAGENTA			0:		YELLOW		
5:		GREEN			A:		WHITE		
6:		ORANGE			B:		LTGRAY		
7:		BROWN			C:		MEDGRAY		
8:		NAVY			D:		GRAY		
9↓		LTBLUE			E:		DARKGRAY		

**CouleurGraph**( $n^{\circ}$ fonction,  $n^{\circ}$ couleur)

Par exemple, **CouleurGraph(2, 4)** ou **CouleurGraph(2, MAGENTA)**.

### OpenLib(

Extension de TI-Basic (non disponible)

### ExeLib(

Extension de TI-Basic (non disponible)

# Instructions PRGM E/S (Entrée/Sortie)

Cette section les instructions **PRGM E/S** (Entrée/Sortie).

## Menu PRGM E/S

Pour afficher le menu **PRGM E/S** (programme entrée/sortie), appuyez sur **PRGM** ► seulement depuis l'éditeur de programme.

**Astuce importante** : Pour retrouver rapidement une commande, utilisez **alpha**  or **alpha**  pour vous déplacer entre les écrans.



**CTRL** **E/S** **COULEUR EXEC HUB**

	<b>Description</b>
1: Input	Entre une valeur ou utilise le curseur.
2: Prompt	Invite pour l'entrée de valeurs de variable.
3: Disp	Affiche le texte, la valeur ou l'écran de calcul.
4: DispGraph	Affiche le graphique courant.
5: DispTable	Affiche le tableau de valeurs courant.
6: Output(	Affiche le texte à une position spécifiée.
7: getKey	Contrôle la frappe sur le clavier.
8: EffÉcran	Efface l'affichage.
9: EffTable	Efface le tableau courant.
0: GetCalc(	Obtient une variable à partir d'une autre TI84 Plus
A: Get(	Obtient une variable à partir d'un dispositif USB connecté. Le dispositif utilisé détermine le mode d'utilisation de cette fonction.

B: Send(	Permet d'envoyer le contenu d'une variable à un dispositif USB connecté. Le dispositif utilisé détermine le mode d'utilisation de cette fonction.
C: eval(	Retourne une expression évaluée sous forme de chaîne avec 8 chiffres significatifs.
D: expr(	Convertit la chaîne de caractères contenue dans <i>chaîne</i> en une expression et l'évalue.
E: toString(	Convertit une valeur en une chaîne de caractères où <i>valeur</i> peut être un réel, un complexe, une expression évaluée, une liste ou une matrice.
F: Chaîne→Équ(	.NEW

**Remarque :** Appuyez sur  $\boxed{+}$  lorsqu'une commande est surlignée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

Ces instructions contrôlent l'entrée et la sortie depuis un programme pendant l'exécution. Elles vous permettent d'entrer des valeurs et affiche les réponses pendant l'exécution du programme.

Pour revenir à l'écran précédent sans sélectionner d'élément, appuyez sur  $\boxed{\text{CLEAR}}$ .

### Affichage d'un graphique avec Input

**Input** sans variable affiche le graphique courant. Vous pouvez déplacer le curseur libre, qui met à jour X et Y (et R et  $\theta$  pour le format **PolarGC**). L'indicateur pause est activé. Appuyez sur  $\boxed{\text{ENTER}}$  pour reprendre l'exécution du programme.

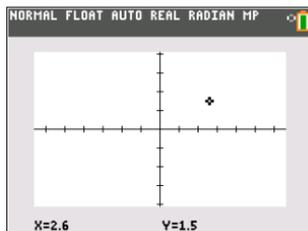
#### Input

##### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:GINPUT
:FnOff
:ZDecimal
:Input
:Disp X,Y
:
```

##### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrmGINPUT
```



## Programme

## Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmGINPUT
                                     2.6
                                     1.5
.....Done.
```

---

## Stocker une Valeur de variable avec Input

**Input** avec *variable* affiche une invite ? (point d'interrogation) pendant l'exécution. *variable* peut être un nombre réel, un nombre complexe, une liste, une matrice, une chaîne de caractères ou une fonction  $Y=$ . Pendant l'exécution du programme, entrez une valeur, pouvant être une expression, puis appuyez sur **ENTER**. La valeur est évaluée et stockée dans *variable* et le programme reprend l'exécution.

---

### Input [*variable*]

Vous pouvez afficher *text* ou le contenu de *Strn* (une variable de type chaîne) jusqu'à 26 caractères comme invite. Pendant l'exécution du programme, entrez une valeur après l'invite, puis appuyez sur **ENTER**. La valeur est stockée dans *variable* et le programme reprend l'exécution.

### Input ["*text*",*variable*]

### Input [*Strn*,*variable*]

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:HINPUT
:Input A
:Input L1
:Input "Y1=",Y1
:Input "DATA=",LDATA
:Disp Y1(A)
:Disp Y1(L1)
:Disp Y1(LDATA)
:■
```

## Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmHINPUT
?2
?{1,2,3}
Y1="2X+2"
DATA={4,5,6}
                                     6
                                     {4 6 8}
{10 12 14}
.....Done.
```

**Remarque :** Lorsqu'un programme invite à entrer des listes et des fonctions  $Yn$  pendant l'exécution, vous devez inclure des accolades ( { } ) autour des éléments de la liste et des guillemets autour ( " ) des expressions.

## Prompt

Pendant l'exécution du programme, **Prompt** affiche chaque *variable*, une à la fois, suivie de =?. À chaque invite, entrez une valeur ou une expression pour chaque *variable*, puis appuyez sur [ENTER]. Les valeurs sont stockées et le programme reprend l'exécution.

**Prompt** *variableA[,variableB,...,variable n]*

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:WINDOW
:Prompt Xmin
:Prompt Xmax
:Prompt Ymin
:Prompt Ymax
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrOmWINDOW
Xmin=?-10
Xmax=?10
Ymin=?-3
Ymax=?3
..... Done.
█
```

**Remarque :** Les fonctions Y= ne sont pas valides avec **Prompt**.

---

## Disp

### Affichage de l'écran d'accueil

**Disp** (affichage) sans valeur affiche l'écran de calcul. Pour voir l'écran de calcul pendant l'exécution du programme, suivez l'instruction **Disp** avec une instruction **Pause**.

### Afficher les valeurs et les messages

**Disp** avec une ou plusieurs *valeurs* affiche la valeur de chacune.

**Disp** [*valeur A,valeur B,valeur C,...,valeur n*]

- Si *valeur* est une variable, la valeur actuelle est affichée.
- Si *valeur* est une expression, elle est évaluée et le résultat est affiché sur le côté droit de la ligne suivante.
- Si *valeur* est un texte entre guillemets, il est affiché sur le côté droit de la ligne d'affichage actuelle. → n'est pas valide comme texte.

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:A
:Disp "THE ANSWER IS ",π/2
```

## Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRGM A
THE ANSWER IS
1.570796327
.....Done.
```

Si **Pause** est rencontré après **Disp**, le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Pour reprendre l'exécution, appuyez sur **ENTER**.

**Remarque** : Si une matrice ou une liste est trop longue pour être affichée entièrement, trois points (...) sont affichés dans la dernière colonne, mais on ne peut pas faire défiler la matrice ou la liste. Pour la faire défiler, utilisez **Pause valeur**.

---

## DispGraph

**DispGraph** (affichage du graphe) affiche le graphique courant. Si **Pause** est rencontré après **DispGraph**, le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Appuyez sur **ENTER** pour reprendre l'exécution.

---

## DispTable

**DispTable** (affichage du tableau) affiche le tableau courant. Le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Appuyez sur **ENTER** pour reprendre l'exécution.

---

## Output(

**Output(** affiche *texte* ou *valeur* sur l'écran d'accueil commençant à *ligne* (1 à 10) et *colonne* (1 à 26), écrasant tout caractère existant.

**Remarque** : Vous voudrez peut-être faire précéder **Output(** de **ClrHome**.

Les expressions sont évaluées et les valeurs sont affichées selon les paramètres du mode courant. Les matrices sont affichées au format d'entrée et retournent à la ligne suivante. → n'est pas valide comme texte.

**Output(ligne,colonne,"texte")**

**Output(ligne,colonne,valeur)**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: OUTPUT
:3+5→B
:ClrDraw
:Output(5,4,"ANSWER: ")
:Output(5,12,B)
:█
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmOUTPUT
..... Done.

ANSWER: 8
```

Pour **Output**( sur un écran partagé **Horiz**, la valeur maximum pour *ligne* est 4.

---

### getKey

**getKey** retourne un chiffre correspondant à la dernière touche pressée, selon le diagramme de code de touches ci-dessous. Si aucune touche n'a été pressée, **getKey** retourne 0. Utilisez **getKey** dans des boucles pour transférer le contrôle, par exemple, lors de la création de jeux vidéo.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: GETKEY
:While 1
:getKey→K
:While K=0
:getKey→K
:End
:Disp K
:If K=105
:Stop
:End█
```

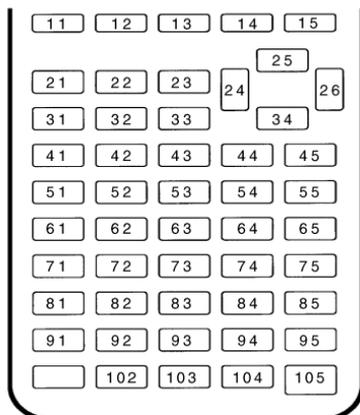
### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmGETKEY
41
42
43
105
..... Done.
```

**Remarque** : **MATH**, **APPS**, **PRGM** et **ENTER** ont été pressée pendant l'exécution du programme.

**Remarque** : Vous pouvez appuyer sur **ON** à tout moment pendant l'exécution pour interrompre le programme.

## Diagramme des codes des touches de la TI-84 Plus



---

### ClrHome, ClrTable

**ClrHome** (effacer l'écran de calcul) efface l'écran de calcul pendant l'exécution du programme.

**ClrTable** (effacer la table de valeurs) efface les valeurs de la table pendant l'exécution du programme.

---

### GetCalc{

**GetCalc{** obtient le contenu de *variable* à partir d'une autre TI-84 Plus CE et le place dans *variable* sur l'unité réceptrice TI-84 Plus CE. *variable* peut être un nombre réel ou complexe, un élément de liste, un nom de liste, un élément de matrice, un nom de matrice, une chaîne de caractères, une variable Y=, une base de données graphique ou une image.

#### GetCalc(*variable*[,*portflag*])

Par défaut, la TI-84 Plus CE utilise le port USB s'il est connecté. Si le câble USB n'est pas connecté, elle utilise le port I/O. Si vous voulez spécifier le port USB ou I/O, utilisez les nombres suivants :

*portflag*=0 utilise le port USB, s'il est connecté ;

*portflag*=1 utilise le port USB ;

*portflag*=2 utilise le port d'E/S. (Ignoré lorsque le programme est exécuté sur la TI-84 Plus CE.)

**Remarque :** **GetCalc()** ne fonctionne pas entre les calculatrices TI-82 et TI-83 Plus ou TI-82, TI-84 Plus et la TI-84 Plus CE.

---

## Get(), Send()

-----

### Get()

**Get()** retrouve une valeur depuis un TI-Innovator™ Hub connecté et stocke les données dans une variable sur la calculatrice CE réceptrice.

### Get(variable)

#### Remarques :

- Utilisez **GetCalc()** pour obtenir des données depuis une autre calculatrice CE.
- Vous pouvez accéder à **Get()**, **Send()** et **GetCalc()** depuis le CATALOG pour les exécuter depuis l'écran d'accueil.

#### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT
:Send("READ BRIGHTNESS ")
:Get(Str1)
:Disp "BRIGHT=",Str1
:■
```

#### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmBRIGHT
BRIGHT=
19.984129
..... Done.
```

#### Astuces TI-Innovator™ Hub :

La définition de la commande **Get()** est spécifique à la calculatrice TI-8x et le câble de connexion via DBus ou USB. La calculatrice CE est à connectivité USB uniquement et ici, **Get()** est destiné à la communication avec le TI-Innovator™ Hub.

Voir également [Send\(\)](#) et [eval\(\)](#).

Voir le [menu HUB](#) pour des détails sur TI-Innovator™ Hub.

-----

### Send()

Envoie une ou plusieurs commandes TI-Innovator™ Hub vers un hub connecté.

### Send(string)

#### Programme

#### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SETCOLOR
:Send("SET COLOR.BLUE ON
TIME 5")
:
```

Allume la LED bleue pendant 5 secondes lorsqu'elle est envoyée au TI-Innovator™ Hub connecté.

### Astuces TI-Innovator™ Hub :

Voir également les commandes [eval\(\)](#) et [Get\(\)](#) liées à la commande **Send()**.

Les commandes TI-Innovator™ Hub sont prises en charge dans le sous-menu HUB dans l'éditeur de programme CE OS v.5.2.

Voir le [menu HUB](#) pour des détails sur TI-Innovator™ Hub.

---

## eval()

eval( retourne une expression évaluée comme une chaîne avec 8 chiffres significatifs. L'expression doit se simplifier à une expression réelle.

eval(expression)

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: EVALHOME
: 5→A
: eval(2A+7)
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mEVALHOME
17.
█
```

### Astuces TI-Innovator™ Hub :

**eval(** peut être utilisé dans une ligne de la commande **Send(**. L'expression évaluée remplace *eval(expression)* avec le résultat comme caractères dans la chaîne

À des fins de débogage, l'utilisation de la ligne de commande **Disp Ans** immédiatement après une ligne de commande utilisant **Send(** affiche l'ensemble de la ligne envoyée.

Voir le [menu HUB](#) pour des détails sur TI-Innovator™ Hub.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SONG2
: {260, 262, 294, 262}→L1
: {4, 4, 2, 2}→L2
: 0→K: 1→T
: For(I, 1, dim(L1))
: Send("SET SOUND eval(2^(K
/12)*L1(I)) TIME eval(T/L2
(I))")
: Disp Ans: Wait T/L2(I)+.05
: End█
```

### Output : Utiliser Disp Ans après la ligne de commande Send(.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mSONG2
SET SOUND 260 TIME 0.25
SET SOUND 262 TIME 0.25
SET SOUND 294 TIME 0.5
SET SOUND 262 TIME 0.5
..... Done
█
```

## expr(

Convertit la chaîne de caractères contenue dans *chaîne* en une expression et l'exécute. *chaîne* peut être une chaîne ou une variable de chaîne.

**expr(chaîne)**

### Programme

```
NORMAL FLOAT DEC REAL RADIAN MP
PROGRAM: EXPR
: 2→X
: "5X"→Str1
: Disp Str1
: expr(Str1)→A
: Disp "A=", A
: █
```

### Sortie

```
NORMAL FLOAT DEC REAL RADIAN MP
PRgmEXPR
5X
A=
..... 10
..... Done
```

---

## toString(

Convertit une valeur en une chaîne où *valeur* peut être réelle, complexe, une expression évaluée, une liste ou une matrice. La chaîne *valeur* s'affiche au *format (0)* classique suivant le paramètre de mode AUTO/DEC ou en *format (1)* décimal.

**toString(valeur[,format])**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TOSTR
: 1/2→A
: Disp toString(A²+2)
: Disp toString(A²+2,0)
: Disp toString(A²+2,1)
: █
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmTOSTR
9/4
9/4
2.25
..... Done
```

## Chaîne→Équ(

**String→Equ(** convertit *string* en une équation et stocke l'équation dans *Yn*. *string* peut être une chaîne ou une variable de type chaîne. **String→Equ(** est l'inverse de **Equ→String(**.

**String→Equ(string,Yn)**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:STREQU
:"2X"→Str2
:String→Equ(Str2,Y2)
:Disp "Y2(-10)=",Y2(-10)
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmSTREQU
Y2(-10)=
-20
.....Done
█
```

## Instructions PRGM COULEUR

Cette section décrit le menu **COULEUR** et les numéros de couleur à utiliser comme arguments lorsqu'une option propose de définir la couleur comme **GraphColor{**.

Vous pouvez copier le jeton de couleur, comme **BLEU**, ou utiliser le numéro de la couleur, comme **10**, affiché dans le tableau ci-dessous.

### Menu PRGM COULEUR

Pour afficher le menu **PRGM COULEUR**, appuyez sur **[PRGM]** **[>]** seulement depuis l'éditeur de programme.

**CTRL**    **E/S**    **COULEUR**    **EXEC**    **HUB**

		Description
1:	BLEU	n°couleur : 10
2:	ROUGE	n°couleur : 11
3:	NOIR	n°couleur : 12
4:	MAGENTA	n°couleur : 13
5:	VERT	n°couleur : 14
6:	ORANGE	n°couleur : 15
7:	MARRON	n°couleur : 16
8:	BLEU MRN	n°couleur : 17
9:	BLEU CLR	n°couleur : 18
0:	JAUNE	n°couleur : 19
A:	BLANC	n°couleur : 20
B:	GRIS CLR	n°couleur : 21
C:	GRIS MOY	n°couleur : 22
D:	GRIS	n°couleur : 23
E:	GRIS FON	n°couleur : 24

**Remarque :** Vous pouvez également choisir un nom dans le menu **[vars]** (sous-menu **COULEUR**).

CTL	I/O	COLOR	EXEC	HUB
1:		BLUE		
2:		RED		
3:		BLACK		
4:		MAGENTA		
5:		GREEN		
6:		ORANGE		
7:		BROWN		
8:		NAVY		
9↓		LTBLUE		

CTL	I/O	COLOR	EXEC	HUB
7↑		BROWN		
8:		NAVY		
9:		LTBLUE		
0:		YELLOW		
A:		WHITE		
B:		LTGRAY		
C:		MEDGRAY		
D:		GRAY		
E:		DARKGRAY		

# Instructions PRGM EXÉC

## Appel d'autres programmes comme sous-programmes

Sur la TI-84 Plus CE, tout programme stocké peut être appelé à partir d'un autre programme comme sous-programme. Entrez le nom du programme à utiliser comme sous-programme seul sur une ligne

## Appel d'un programme à partir d'un autre programme

Vous pouvez entrer le nom d'un programme sur une ligne de commande des deux façons.

- Appuyez sur **PRGM** **[4]** pour afficher le menu **PRGM EXEC** et sélectionner le nom du programme **prgmnom** est collé sur l'emplacement actuel du curseur sur une ligne de commande.
- Sélectionnez **prgm** depuis le menu **PRGM CTL**, puis entrez le nom du programme.

**prgmnom**

Lorsque **prgmnom** est rencontré pendant l'exécution, la prochaine commande que le programme exécute est la première commande dans le deuxième programme. Il retourne à la commande suivante du premier programme lorsqu'il rencontre soit **Return** soit le **Return** implicite à la fin du deuxième programme.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: VOLCYL
: Input "D=", D
: Input "H=", H
: PRGMAREACIR
: A*H→V
: DISP V
: █
```



### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRGMVOLCYL
D=4
H=5
62.83185307
Done
█
```

### Sous-programme ↓↑

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: AREACIR
: D/2→R
: π*R²→A
: Return
```

### Remarques à propos de l'appel de programmes

Les variables sont globales.

*label* utilisé avec **Goto** et **Lbl** est local au programme où il est situé. *label* dans un programme n'est pas reconnu par un autre programme. Vous ne pouvez pas utiliser **Goto** pour brancher vers un *label* dans un autre programme.

**Return** sort d'un sous-programme et retourne au programme appelant, même s'il est rencontré à l'intérieur d'une boucle.

# Instructions PRGM HUB

## Instructions du Menu TI-Innovator™ HUB

Cette section décrit les instructions du menu TI-Innovator™ HUB.

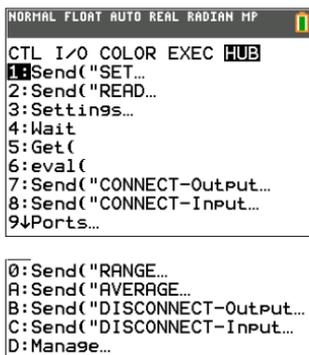
Voir les activités [TI-Innovator™ System](#) pour les détails et les valeurs des paramètres pour les capteurs et les contrôles spécifiques. Cette section décrit les instructions ou les commandes contenues dans le menu TI-Innovator™ HUB et la façon dont les commandes sont copiées dans l'éditeur de programme.

## Menu TI-Innovator™ HUB

Pour afficher le menu HUB TI-Innovator™, appuyez sur **PRGM** seulement depuis l'éditeur de programme.

**Astuce importante :** Pour retrouver rapidement une commande, utilisez **[alpha]** **▲** or **[alpha]** **▼** pour vous déplacer entre les écrans.

- Si **[A-lock]** est **on**, alors **▲** et **▼** fera défiler les écrans des menus et l'écran d'édition de programme.
- Après avoir entré des caractères alphabétiques, pensez à **turn off** **[A-lock]** pour éviter la pagination imprévue des écrans.



**Remarque :** Toute commande TI-Innovator™ Hub peut également être entrée caractère par caractère.

Les commandes TI-Basic comme **Send()**, **Get()**, **Wait** et **eval()** doivent être copiées comme jetons depuis les menus.

**CTRL**    **E/S**    **COULEUR EXEC**    **HUB**

	Description
1: Send("SET...	Construit une commande <b>Send()</b> à coller dans l'éditeur
2: Send("READ...	Construit une commande <b>Send()</b> à coller dans l'éditeur
3: Paramètres...	Copie une commande TI-Innovator™ Hub dans l'éditeur
4: Wait	Copie une commande TI-Basic dans l'éditeur
5: Get(	Copie une commande TI-Basic dans l'éditeur
6: eval(	Copie une commande TI-Basic dans l'éditeur
7: Send("CONNECT-OUTPUT	Construit une commande <b>Send()</b> à coller dans l'éditeur

8:	Send("CONNECT-INPUT	Construit une commande <b>Send()</b> à coller dans l'éditeur
9:	Port :	Copie une commande TI-Innovator™ Hub dans l'éditeur
0:	Send("RANGE...	Construit une commande <b>Send()</b> à coller dans l'éditeur
A:	Send("AVERAGE...	Construit une commande <b>Send()</b> à coller dans l'éditeur
B:	Send("DISCONNECT-OUTPUT	Construit une commande <b>Send()</b> à coller dans l'éditeur
C:	Send("DISCONNECT-INPUT	Construit une commande <b>Send()</b> à coller dans l'éditeur
D:	Manage...	Copie plusieurs commandes (:) dans l'éditeur

Pour obtenir l'aide du Catalogue lors de l'utilisation des commandes **eval()**, **Get()**, ou **Wait**, appuyez sur **[+]**.

Pour retourner vers l'éditeur de programme sans sélectionner d'élément, appuyez sur **[CLEAR]** jusqu'à ce que le curseur retourne à l'éditeur de programme.

**Attention :** N'appuyez pas sur **[CLEAR]** plusieurs fois, à moins que vous ne soyez sur la navigation de l'écran. Une fois les menus effacés, si vous appuyez sur effacer à nouveau, vous risquez d'effacer une ligne entière de votre programme. Il n'y a pas de fonction d'annulation dans l'éditeur de programme.

---

## Avant de commencer

Cette section fournit des descriptions sur la façon dont chaque élément de menu est copié dans l'éditeur de programme. Pour des informations spécifiques sur TI-Innovator™ System concernant la syntaxe et les paramètres de chaque capteur, consultez les informations spécifiques dans les activités et les kits [TI-Innovator™ System](#). Vous pouvez également entrer la commande TI-Innovator™ Hub (uniquement) lettre par lettre, en utilisant la touche [alpha], [ " ], [ \_ ], etc.

Les espaces supplémentaires sont collés par souci de facilité. Le schéma TI-Innovator™ Hub ignorera les espaces supplémentaires avec des guillemets dans une commande **Send()**. Cependant, lorsque vous exécutez votre programme, les espaces supplémentaires ne peuvent pas se trouver à la fin des lignes de commande, sinon ils donneraient une erreur de syntaxe. Si vous avez une erreur de syntaxe à la fin d'une ligne, cherchez les espaces supplémentaires et supprimez-les.

### **Comment la commande *Send()* construit-elle une commande TI-Innovator™ Hub depuis le menu HUB.**

Depuis le menu **HUB**, sélectionnez une commande **Send()**. L'écran suivant vous donnera des options pour ce format de **Send()**.

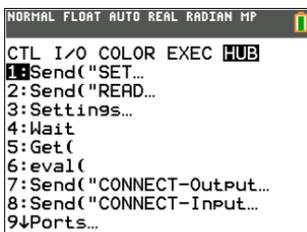
**Exemple :** Pour copier **Send()**"SET COLOR.RED dans l'éditeur de programme, suivez ces étapes.

1. Avec le curseur sur une ligne de commande dans l'éditeur de programme, appuyez sur **[prgm]** pour accéder aux menus de commande de programmation.



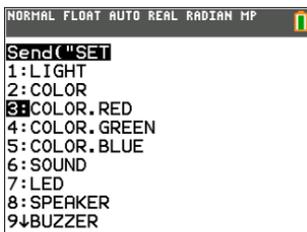
```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:SET
:█
```

2. Appuyez sur **[4]** pour sélectionner le menu **HUB**.  
Sélectionnez **1:Send("SET...**  
Les "..." indiquent qu'il existe un autre menu d'options.



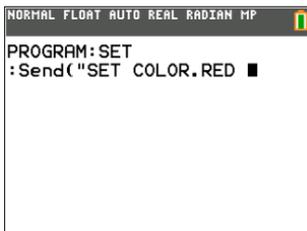
```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:Send("SET...
2:Send("READ...
3:Settings...
4:Wait
5:Get(
6:eval(
7:Send("CONNECT-Output...
8:Send("CONNECT-Input...
9↓Ports...
```

3. Sélectionnez **3:COLOR.RED**.



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("SET
1:LIGHT
2:COLOR
3:COLOR.RED
4:COLOR.GREEN
5:COLOR.BLUE
6:SOUND
7:LED
8:SPEAKER
9↓BUZZER
```

4. L'entière ligne de commande **Send(** est copiée dans l'éditeur de programme. Répétez l'opération pour sélectionner plus de commandes TI-Innovator™ Hub. Utilisez **[alpha]** **["]** et **[ ) ]** pour terminer la commande **Send(** lorsque ceci est nécessaire.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:SET
:Send("SET COLOR.RED █
```

**Remarque :** Toute commande TI-Innovator™ Hub utilisant **Send(** entre guillemets peut être entrée en utilisant les touches [alpha] sur le clavier.

Pour les couleurs, n'utilisez pas la commande du jeton COULEUR depuis le menu COULEUR lorsque vous communiquez avec TI-Innovator.™ Hub.

## Send("Set...

La commande **SET** indique au schéma "TI-Innovator™ Hub de **SET** la valeur de l'objet spécifié. Elle supporte tous les objets 'nommés'.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("SET
1: LIGHT
2: COLOR
3: COLOR. RED
4: COLOR. GREEN
5: COLOR. BLUE
6: SOUND
7: LED
8: SPEAKER
9: BUZZER
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("SET
0: RELAY
A: SERVO
B: SERVO. CONTINUOUS
C: DC MOTOR
D: SQUAREWAVE
E: RGB
F: ANALOG. OUT
G: DIGITAL. OUT
H: AVERAGING
```

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SET
: Send("SET COLOR. RED ON)
: Wait 1.5
: Send("SET COLOR. RED OFF)
:
```

## Sortie

**Exemple** : Ceci allume une **LED** rouge pendant 1,5 seconde puis l'éteint.

**Remarque** : La **LED** d'alimentation est verte.

Les commandes **ON** et **OFF** peuvent être entrées ou trouvées dans l'élément de menu **Settings...** dans le menu **HUB**.

Utilisez [alpha] [ \_ ] pour les espaces si nécessaire.

## Send("READ...

La commande **READ** doit indiquer au schéma "TI-Innovator™ de SET la valeur du port/pin/objet spécifié. Elle supporte tous les objets 'nommés'. Elle peut également être utilisée avec des adresses pin 'brutes'. Elle doit être suivie d'une commande **Get** pour transférer les informations vers une variable pour utiliser ou afficher la valeur de la variable.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("READ
1: BRIGHTNESS
2: DHT
3: RANGER
4: LOUDNESS
5: LIGHTLEVEL
6: TEMPERATURE
7: BUTTON
8: MOTION
9↓ POTENTIOMETER
```

```
0: POTENTIOMETER
A: MOISTURE
B: THERMISTOR
C: ANALOG.IN
D: DIGITAL.IN
E: AVERAGING
```

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: READ
: Send("READ BRIGHTNESS)
: Get(B)
: Disp B
: █
```

## Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmREAD
10.974791
.....Done
█
```

## Paramètres...

Le menu **Settings** contient des opérations pour paramétrer l'état des opérations de pin numériques ou analogiques comme la **LED** dans le TI-Innovator™ Hub ou un mouvement de servomoteur vers les états comme **ON**, **OFF**, **CW** (sens horaire) et **CCW** (sens antihoraire).

Voir les kits d'activité [TI-Innovator™ System](#) pour plus de détails.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Settings
1: ON
2: OFF
3: TO
4: TIME
5: BLINK
6: TEMPERATURE
7: HUMIDITY
8: CW
9↓CCW
```

```
0: TOGGLE
A: PULLUP
B: PULLDOWN
C: INPUT
```

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SET
:Send( "SET COLOR.RED ON)
:Wait 1.5
:Send( "SET COLOR.RED OFF)
:
```

## Sortie

**Exemple** : Ceci allume une **LED** pendant 1,5 seconde puis l'éteint.

**Rappel** : La **LED** d'alimentation est verte.

## Wait

**Wait** suspend l'exécution d'un programme pour un temps donné. Le temps maximum donné est 100 secondes. Pendant le temps de pause, l'indicateur d'occupation est situé dans le coin supérieur droit.

Attendez *time*

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:WAIT
:ClrDraw
:AxesOff:F0ff
:TextColor(MAGENTA)
:Text(2,2,"HELLO WORLD"
:Wait 5
:TextColor(GREEN)
:Text(24,2,"BYE!"
:
```

### Output : "Bye!" s'affiche après 5 secondes.

```
NORMAL FLOAT AUTO REAL RADIAN MP
HELLO WORLD
BYE!
```

### Astuces TI-Innovator™ Hub :

**Wait** peut être utilisé dans les programmes TI-Innovator™ Hub pour laisser du temps aux communications de capteur ou de contrôle avant l'exécution de la ligne de commande suivante par le programme.

## Get(

**Get(** récupère une valeur depuis un TI-Innovator™ Hub connecté et stocke les données dans une variable sur la calculatrice CE réceptrice.

**Get(variable)**

### Remarques :

- Utilisez **GetCalc(** pour obtenir des données depuis une autre calculatrice CE.
- Vous pouvez accéder à **Get(**, **Send(** et **GetCalc(** depuis le CATALOG pour les exécuter depuis l'écran de calcul.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT
:Send("READ BRIGHTNESS ")
:Get(Str1)
:Disp "BRIGHT=",Str1
:■
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mBRIGHT
BRIGHT=
19.984129
..... Done.
```

### Astuces TI-Innovator™ Hub :

La définition de la commande **Get(** est spécifique à la calculatrice TI-8x et au câble de connexion via DBus ou USB. La calculatrice CE est à connectivité USB uniquement et ici, **Get(** est destiné à la communication avec le TI-Innovator™ Hub.

Voir également [Send\(](#) et [eval\(](#).

## eval()

**eval()** retourne une expression évaluée comme chaîne avec 8 chiffres significatifs. L'expression doit se simplifier à une expression réelle.

**eval(expression)**

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: EVALHOME
:5→A
:eval(2A+7)
:
```

### Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmEVALHOME
17.
█
```

### Astuces TI-Innovator™ Hub :

**eval()** peut être utilisé dans une ligne de la commande **Send()**. L'*expression* évaluée remplace **eval(expression)** avec le résultat comme caractères dans la chaîne

À des fins de débogage, l'utilisation de la ligne de commande **Disp Ans** immédiatement après une ligne de commande utilisant **Send()** affiche l'entière chaîne de caractères envoyée.

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SONG2
:{260,262,294,262}→L1
:{4,4,2,2}→L2
:0→K:1→T
:For(I,1,dim(L1))
:Send("SET SOUND eval(2^(K
/12)*L1(I)) TIME eval(T/L2
(I))")
:Disp Ans:Wait T/L2(I)+.05
:End█
```

### Output : Utiliser Disp Ans après la ligne de commande Send().

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmSONG2
SET SOUND 260 TIME 0.25
SET SOUND 262 TIME 0.25
SET SOUND 294 TIME 0.5
SET SOUND 262 TIME 0.5
.....Done.
█
```

## Send("CONNECT-OUTPUT...

CONNECT (Output) associe un contrôle ou un capteur donné à un pin ou un port sur le TI-Innovator™.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("CONNECT
1: LIGHT
2: COLOR
3: SOUND
4: LED
5: SPEAKER
6: BUZZER
7: RELAY
8: SERVO
9↓SERVO.CONTINUOUS

0: DCMOTOR
A: SQUAREWAVE
B: RGB
C: ANALOG.OUT
D: DIGITAL.OUT
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CONNECT
:Send("BEGIN")
:Wait 1
:Send("CONNECT SERVO 1 TO
OUT3")
:
```

### Sortie

Connecte le servomoteur à OUT3.

## Send("CONNECT-INPUT...

CONNECT (Input) associe un contrôle ou un capteur donné à un pin ou un port sur le TI-Innovator™.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("CONNECT
1: LIGHT
2: COLOR
3: SOUND
4: LED
5: SPEAKER
6: BUZZER
7: RELAY
8: SERVO
9↓SERVO.CONTINUOUS

0: DCMOTOR
A: SQUAREWAVE
B: RGB
C: ANALOG.OUT
D: DIGITAL.OUT
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CONECTIN
:Send("BEGIN")
:Wait 0.5
:Send("CONNECT RANGER 1 TO
IN 1)
:
```

### Sortie

Connecte un télémètre externe à IN 1.

## Ports...

Le menu Ports liste les ports disponibles à connecter comme entrée, sortie ou à une platine d'expérimentation.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Ports
1:OUT 1
2:OUT 2
3:OUT 3
4:IN 1
5:IN 2
6:IN 3
7:I2C
8:BB 1
9↓BB 2

0:BB 3
A:BB 4
B:BB 5
C:BB 6
D:BB 7
E:BB 8
F:BB 9
G:BB 10
```

## Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CONECTIN
:Send("BEGIN")
:Wait 0.5
:Send("CONNECT RANGER 1 TO
IN 1)
:
```

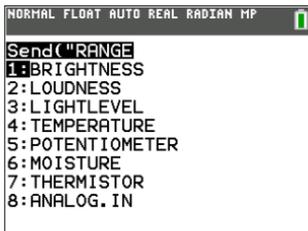
## Sortie

Connecte un télémètre externe à IN 1.

**Remarque :** Les espaces supplémentaires sont copiés comme l'espace dans "IN 1" "IN1" est également accepté par le schéma de TI-Innovator™ Hub sur TI-Innovator™.

## Send("RANGE...

Change ou définit la plage à une plage choisie par l'utilisateur d'une valeur minimum à une valeur maximum.



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("RANGE
1: BRIGHTNESS
2: LOUDNESS
3: LIGHTLEVEL
4: TEMPERATURE
5: POTENTIOMETER
6: MOISTURE
7: THERMISTOR
8: ANALOG. IN
```

### Exemples de Syntaxe :

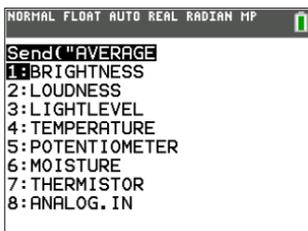
```
Send("RANGE BRIGHTNESS minimum maximum")
```

```
Send("RANGE LIGHTLEVEL # minimum maximum")
```

---

## Send("AVERAGE...

La commande AVERAGE est utilisée pour définir le nombre d'échantillons pris pour représenter la lecture moyenne d'un capteur.



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("AVERAGE
1: BRIGHTNESS
2: LOUDNESS
3: LIGHTLEVEL
4: TEMPERATURE
5: POTENTIOMETER
6: MOISTURE
7: THERMISTOR
8: ANALOG. IN
```

### Exemples de Syntaxe :

```
Send("AVERAGE BRIGHTNESS number")
```

```
Send("AVERAGE LIGHTLEVEL # number")
```

Où "number" est le nombre de lectures dont la moyenne doit être faite.

## Send("DISCONNECT-OUTPUT...

CONNECT (Output) rompt l'association entre un contrôle ou un capteur spécifique et un pin ou un port sur le TI-Innovator™.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("DISCONNECT
1: LIGHT
2: COLOR
3: SOUND
4: LED
5: SPEAKER
6: BUZZER
7: RELAY
8: SERVO
9↓SERVO.CONTINUOUS
```

```
0: DCMOTOR
A: SQUAREWAVE
B: RGB
C: ANALOG.OUT
D: DIGITAL.OUT
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: DISCOUT
: Send( "BEGIN" )
: Send( "DISCONNECT COLOR" )
: █
```

### Sortie

Déconnecte l'utilisation de la LED RGB intégrée.

## Send("DISCONNECT-Input...

CONNECT (Input) rompt l'association entre un contrôle ou un capteur spécifique et un pin ou un port sur le TI-Innovator™.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("DISCONNECT
1:BRIGHTNESS
2:DHT
3:RANGER
4:LOUDNESS
5:LIGHTLEVEL
6:TEMPERATURE
7:SWITCH
8:BUTTON
9↓MOTION
```

```
0:POTENTIOMETER
A:MOISTURE
B:THERMISTOR
C:ANALOG.IN
D:DIGITAL.IN
```

### Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:PROG05
:Send("BEGIN")
:Wait 0.5
:Send("DISCONNECT RANGER 1
")
:█
:
```

### Sortie

Déconnecte l'utilisation du capteur de plage.

## Manage...

Le menu **Manage** copie une commande **Send**( avec les éléments de gestion suivants. **Str0** est affiché dans l'écran de calcul avec les informations, si elles sont demandées dans la commande.

- **BEGIN** – déconnecte tous les capteurs et contrôles connectés. **Send("BEGIN")** peut être nécessaire dans un programme TI-Innovator™ Hub pour réinitialiser un capteur ou un contrôle avant d'envoyer une commande vers ce capteur ou ce contrôle.
- **ISTI** – Répond avec TI STEM
- **WHO** – Répond avec TI-Innovator™ Hub ON MSP432
- **WHAT** – Répond avec TI-Innovator™ Hub
- **HELP** – Répond avec USE HELP COMMAND FOR DETAILS
- **VERSION** – Répond avec le numéro de version TI-Innovator™ Hub
- **ABOUT** – Répond avec TI-Innovator™ Hub ©2016 Texas Instruments

```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("
1:BEGIN"):Get(Str0):Disp
2:ISTI"):Get(Str0):Disp
3:WHO"):Get(Str0):Disp
4:WHAT"):Get(Str0):Disp
5:HELP"):Get(Str0):Disp
6:VERSION"):Get(Str0):Disp
7:ABOUT"):Get(Str0):Pause
```

```
0:POTENTIOMETER
A:MOISTURE
B:THERMISTOR
C:ANALOG.IN
D:DIGITAL.IN
```

**Remarque** : Le [ : ] est utilisé pour séparer les lignes de commande sur une seule ligne de commande. Le menu **Manage**... colle un ensemble pratique de commandes pour ensuite afficher les informations dans **Str0** sur l'écran de calcul.

## Exécution d'un programme en assembleur

Vous pouvez utiliser des programmes écrits pour la TI-84 Plus CE en assembleur. En général, les programmes en assembleur s'exécutent plus rapidement et offre un plus grand contrôle que les programmes utilisant des combinaisons de touches que vous créez dans l'éditeur de programmes intégré.

**Remarque :** Un programme en assembleur ayant un plus grand contrôle sur la calculatrice, si votre programme en assembleur contient une ou plusieurs erreurs, il peut entraîner le redémarrage de la calculatrice et la perte de toutes les données, programmes et applications stockées dans votre mémoire.

Lorsque vous téléchargez un programme en assembleur, il est stocké avec les autres programmes comme élément de menu **PRGM**. Vous pouvez effectuer les opérations suivantes :

- Le transmettre à l'aide du câble de communication de la TI-84 Plus CE
- Le supprimer à l'aide de l'écran Memory Management/ Delete

Pour exécuter un programme en assembleur, la syntaxe est : **Asm**  
(*assemblyprgmname*)

Si vous écrivez un programme en assembleur, utilisez les deux instructions ci-dessous depuis le CATALOG pour identifier et compiler le programme.

Instructions	Commentaires
<b>AsmComp</b> ( <i>prgmASM1</i> , <i>prgmASM2</i> )	Compile un programme en assembleur écrit en ASCII et stocke la version hexadécimale.
<b>Asm84CEPrgm</b>	Identifie un programme en assembleur, doit être entré comme la première ligne d'un programme en assembleur.

Pour compiler un programme en assembleur que vous avez écrit :

1. Suivez les étapes pour écrire un programme (16-4) mais assurez-vous d'inclure **Asm84CEPrgm** comme première ligne de votre programme.
2. Depuis l'écran de calcul, appuyez sur  $\boxed{2\text{nd}}$  [CATALOG] et sélectionnez **AsmComp** pour le coller sur l'écran.
3. Appuyez sur  $\boxed{\text{PRGM}}$  pour afficher le menu **PRGM EXEC**.
4. Sélectionnez le programme à compiler. Il sera copié dans l'écran de calcul.
5. Appuyez sur  $\boxed{.}$  et sélectionnez **prgm** depuis le **CATALOG**.
6. Touche du nom que vous avez choisi pour le programme de sortie.

**Remarque :** Ce nom doit être unique — pas une copie du nom d'un programme existant.

7. Appuyez sur  $\boxed{\square}$  pour terminer la séquence.

La séquence des arguments doit être comme suit :

**AsmComp**(*prgmASM1*, *prgmASM2*)

8. Appuyez sur **ENTER** pour compiler votre programme et générer le programme de sortie.

**Remarque** : La TI-84 Plus ou TI-84 Plus C Silver Edition **AsmPrgm transfère vers TI-84 Plus CE mais échoue à l'exécution.**

# Informations générales

## **Support et service de Texas Instruments**

### Informations générales : Amérique du Nord et du Sud

Page d'accueil :	<a href="http://education.ti.com">education.ti.com</a>
Base de connaissances et demandes par e-mail :	<a href="http://education.ti.com/support">education.ti.com/support</a>
Téléphone :	(800) TI-CARES / (800) 842-2737 Pour l'Amérique du Nord et du Sud et les territoires américains

Informations pour les contacts internationaux : [education.ti.com/support/worldwide](http://education.ti.com/support/worldwide)

### Assistance produit

Base de connaissances et assistance par e-mail :	<a href="http://education.ti.com/support">education.ti.com/support</a> ou <a href="mailto:ti-cares@ti.com">ti-cares@ti.com</a>
Téléphone (numéro payant) :	(972) 917-8324

### Pour les services produit (matériel)

**Clients aux États-Unis, au Canada, au Mexique et dans les territoires américains :**  
Contactez toujours le centre de support clientèle de Texas Instruments avant de renvoyer un produit en réparation

### Pour tous les autres pays :

#### Informations générales

Pour plus d'informations concernant les produits et services TI, contactez TI par e-mail ou visitez le site Web de TI

Demandes par e-mail :	<a href="mailto:ti-cares@ti.com">ti-cares@ti.com</a>
Page d'accueil :	<a href="http://education.ti.com">education.ti.com</a>

## **Informations Garantie et Assistance**

Pour de plus amples informations concernant la durée et les conditions de la présente garantie ou le support d'un produit, reportez-vous à la Carte de garantie qui accompagne ce produit ou contactez votre revendeur/distributeur Texas Instruments local

# Index

(			
(: (deux-points) .....	12		
<b>A</b>			
Asm( .....	61		
AsmComp( .....	61		
<b>C</b>			
CBL 2™ .....	37		
CBR™ .....	37		
Compilation d'un programme en assembleur .....	61		
CouleurGraph .....	28		
<b>D</b>			
deux-points séparateur (:) .....	12		
diagramme code-touche .....	36		
Disp (afficher) .....	33		
DispGraph (afficher graphe) .....	34		
DispTable (afficher table de valeurs) ..	34		
<b>E</b>			
effacer			
table (EffTable) .....	36		
Effacer			
écran de calcul (EffÉcran) .....	36		
EffÉcran (effacer écran de calcul) ...	36		
EffTable (effacer table) .....	36		
Else .....	19		
End .....	20		
étiquettes			
programme .....	24		
eval( .....	39, 53		
exemples -- Premiers contacts			
volume d'un cylindre .....	3		
expr( .....	40		
<b>F</b>			
For( .....	20		
<b>G</b>			
Get( .....	52		
Get( (obtenir des données du système CBL 2™ ou CBR™) .	37		
GetCalc( (obtenir des données d'une TI84 Plus) .....	36		
getKey .....	35		
Goto .....	24		
GraphStyle( .....	27		
<b>H</b>			
hub			
instructions .....	45		
<b>I</b>			
Incrémenter et ignorer (IS>()) .....	25		
Input .....	31-32		
instructions If			
If .....	18		
If-Then .....	18		
If-Then-Else .....	19		
IS>( (incrémenter et ignorer) .....	25		
<b>L</b>			
Lbl (étiquette) .....	24		
<b>M</b>			
Manage... .....	60		
Menu HUB (menu Hub) .....	45		
menu PRGM EDIT .....	15		
menu PRGM EXEC .....	15		
Menu PRGM NOUVEAU .....	8		
Menu( (définir menu) .....	26		
menus			
définition (Menu()) .....	26		
<b>O</b>			
Output( .....	34		
<b>P</b>			
Pause .....	23		

PGRM CTL (menu contrôle de programme) .....	16	Stop .....	27
Ports... .....	56	String>Equ .....	41
prgm (nom programme) .....	26	supprimer le contenu d'une variable (EffVar) .....	27
PrgmAsm( .....	61		
programmation		<b>T</b>	
augmenter la mémoire disponible .....	9	Then .....	18
copier et renommer .....	15	TI-84 Plus	
créer nouveau .....	8	diagramme des codes des touches .....	36
définition .....	2	toString( .....	40
entrer des lignes de commande	12		
exécution .....	12	<b>W</b>	
exécution d'un programme en assembleur .....	61	Wait .....	24, 51
insérer des lignes de commande	14	While .....	21
instructions .....	16		
interruption .....	13		
modification .....	14		
nom (prgm) .....	26		
renommer .....	15		
sous-programmes .....	43		
suppression .....	9		
supprimer des lignes de commande .....	14		
programmes en assembleur .....	61		
Prompt .....	33		
<b>R</b>			
Repeat .....	21		
Return .....	27		
<b>S</b>			
Send( (envoyer à CBL 2™ ou CBR™) .	37		
Send("AVERAGE... .....	57		
Send("CONNECT-INPUT... .....	55		
Send("CONNECT-OUTPUT... .....	54		
Send("DISCONNECT-INPUT... .....	59		
Send("DISCONNECT-OUTPUT... .....	58		
Send("RANGE... .....	57		
Send("READ... .....	49		
Send("SET... .....	48		
sous-routine .....	26		