Jean-Baptiste Civet, Boris Hanus et Isabelle Pazé

# Prise en main de la calculatrice graphique en mathématiques


**Classes de 2nde (LEGT & LP)**

$T^3$ FRANCE
Teachers Teaching with Technology™
www.t3france.fr

# Activités mathématiques en classe de seconde

## Table des matières

# Mises à jour système et compléments logiciels

Pour réaliser les fiches proposées dans cet ouvrage, il est nécessaire de disposer d'une calculatrice Texas Instruments, permettant également de programmer en langage Python, cela bien que ce ne soit pas l'objectif premier de ce livret.

Cela concerne les modèles suivants :

- TI-82 Advanced Édition Python
- TI-83 Premium CE + Python Adapter
- TI-83 Premium CE Édition Python

Pour les deux derniers modèles, il convient que le système d'exploitation de la calculatrice soit à jour.

L'installation des mises à jour calculatrice est très facile : il suffit de télécharger un fichier à l'adresse indiquée ci-dessous puis de copier le fichier téléchargé dans la machine à l'aide du logiciel ad hoc (à savoir le logiciel TI-Connect CE prévu à cet effet).

| | |
|---|---|
| Mise à jour des calculatrices TI-83 Premium CE | |

# AVANT PROPOS

Ce livret est destiné aux élèves de seconde désireux de prendre en main leur calculatrice, de découvrir l'usage convenable des principales touches et des outils classiques dans le cadre des pratiques habituellement réalisées en cours de Mathématiques.

Les thèmes du calcul numérique, de l'analyse, des statistiques et probabilités ainsi que de la programmation en Python sont abordés.

Il a été décidé d'intégrer, pour chaque thème, d'une part, une progressivité dans les fiches proposées et, d'autre part, les fiches se présentent soit sous la forme de fiches outils synthétisant l'ensemble des manipulations utiles à connaître pour une action donnée (par exemple « savoir représenter des fonctions »), soit sous la forme d'un « exercice type » résolu à l'aide de la calculatrice.

Les notions sont présentées de façon progressive, mais aussi de façon assez indépendante afin de pouvoir utiliser confortablement n'importe quelle fiche. Des rappels (un peu moins détaillés) sont faits pour une notion déjà vue dans le livret, lorsque la manipulation enchaine plusieurs actions. La pédagogie, c'est aussi la répétition !

Ainsi le professeur de mathématiques ou de sciences, pourra photocopier, diffuser la séquence qu'il souhaite aborder avec ses élèves afin que ces derniers puissent avoir une aide précieuse et abordable en seconde.

Les auteurs ont souhaité aborder l'ensemble des notions que tout professeur traite en classe de seconde à l'aide de la calculatrice. L'aspect Python est aussi abordé mais plutôt sous l'angle de notions indispensables qui ne balayent pas tout le bagage que doit maîtriser un élève de seconde.

Le passage du collège au lycée est délicat pour un élève qui doit découvrir une nouvelle calculatrice beaucoup plus complète que celle qu'il avait au collège. D'autre part il doit aussi découvrir le langage Python...

On traite, dans les pages qui suivent, un usage simple et efficace, des calculatrices TI-83 Premium CE Edition Python ainsi que la TI-82 Advanced Edition Python.

Pour plus d'information sur ces produits, vous pouvez consulter https://education.ti.com/fr section « calculatrices lycée et supérieur ».

Vous pouvez utiliser indifféremment l'une ou l'autre de ces machines pour les activités présentes dans ce livret. La différence entre ces deux machines se situe surtout au niveau du langage Python. La TI-82 Advanced Edition Python permet d'utiliser les bibliothèques math et random et la TI-83 Premium CE Edition Python possède des bibliothèques supplémentaires comme ti_plotlib, ti_rover, microbit, et bien d'autres encore.

# Calculs de fractions, racines…

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Touche fraction [▫/▫]

Pour écrire des nombres sous forme de fraction comme sur le cahier, on utilise la touche [▫/▫].

La calculatrice va simplifier la fraction pour l'écrire sous forme irréductible.

Exemple 1 : Si on souhaite simplifier $a = \frac{102}{66}$, on appuie sur [▫/▫]

Puis on entre tout simplement le numérateur,

ainsi que le dénominateur.


Après avoir entré le numérateur et le dénominateur il suffit tout simplement de valider en appuyant sur [entrer]. On trouve donc $a = \frac{17}{11}$.


Exemple 2 : Exprimer sous forme de fraction irréductible $b = \frac{3}{4} + \frac{17}{9} - \frac{1}{5}$

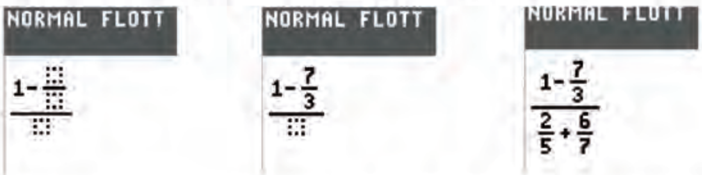
A chaque nouvelle fraction on appuie sur [▫/▫] pour construire $b$ :



On trouve que $b = \frac{439}{180}$.

Exemple 3 : Reprenons le même énoncé avec $c = \frac{1-\frac{7}{3}}{\frac{2}{5}+\frac{6}{7}}$

Créer une fraction (touche [▫/▫]), puis écrire le début du numérateur :





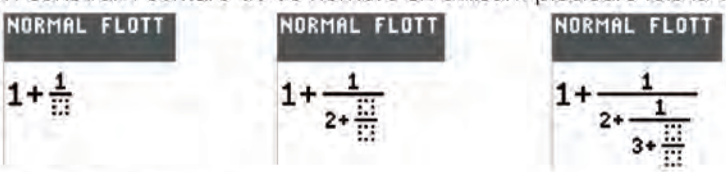
Dès qu'on a besoin de créer une autre fraction, on appuie sur [▫/▫]

On trouve finalement que $c = -\frac{35}{33}$.

On peut enchainer des fractions pour effectuer des calculs plus compliqués !

Exemple 4 : Calculer $d = 1 + \frac{1}{2+\frac{1}{3+\frac{1}{4}}}$

On construit l'écriture de ce nombre en utilisant plusieurs fois la touche [▫/▫]





On trouve ainsi $d = \frac{43}{30}$

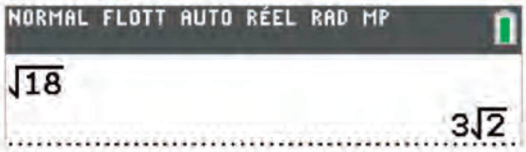
# Calculs de fractions, racines…

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## Touche racine carrée

La calculatrice sait aussi faire des calculs qui utilisent la racine carrée…




Exemple 1 : Simplifier le plus possible le nombre $A = \sqrt{18}$.

Pour obtenir $\sqrt{\phantom{x}}$, appuyer sur 2nde x². Après avoir validé, la calculatrice donne le résultat le plus simplifié possible.

Conclusion : $A = 3\sqrt{2}$.

Exemple 2 : Avec le même énoncé prenons $B = \sqrt{8} + 5\sqrt{50} - 2\sqrt{2}$

On construit l'expression progressivement à l'aide du symbole $\sqrt{\phantom{x}}$ :

Il ne faut pas oublier d'appuyer sur › pour sortir de la racine carrée.

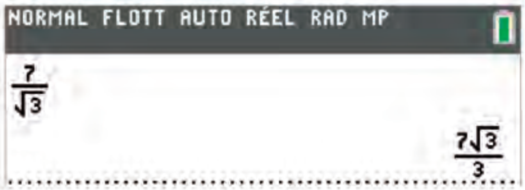


On trouve $b = 25\sqrt{2}$

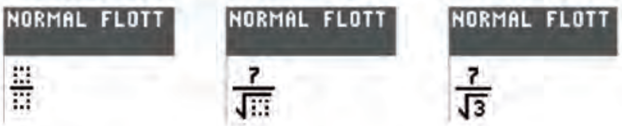
En mathématiques, on demande souvent, lorsqu'on a une fraction, de rendre rationnel le dénominateur (c'est-à-dire « d'enlever » les racines carrées au dénominateur).

Votre calculatrice sait le faire !

Exemple 3 : Rendre rationnel le dénominateur du nombre $C = \frac{7}{\sqrt{3}}$



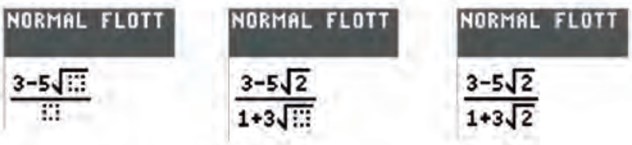

On commence par construire une fraction à l'aide de ⁿ⁄ₐ puis on entre la racine carrée au dénominateur en appuyant sur 2nde x².




On obtient $C = \frac{7\sqrt{3}}{3}$.

Parfois le calcul peut être beaucoup plus long mais la calculatrice va vous permettre de vérifier vos résultats.

Exemple 4 : Rendre rationnel le dénominateur du nombre $D = \frac{3-5\sqrt{2}}{1+3\sqrt{2}}$


On trouve $D = \frac{-33+14\sqrt{2}}{17}$

# Calculs de carrés, de puissances…

TI-82 Advanced Edition Python

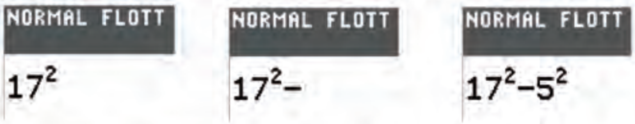
TI-83 Premium CE Edition Python

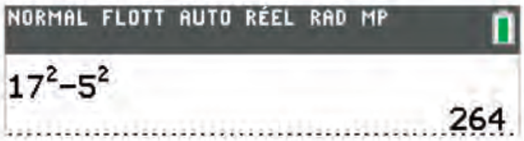
## Touche carré $x^2$

Lorsqu'on souhaite calculer un carré, on dispose d'une touche permettant de faire ce calcul rapidement en utilisant la même notation qu'en mathématiques.

Exemple 1 : Calculer $17^2 - 5^2$

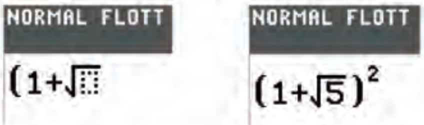
On construit notre expression progressivement :

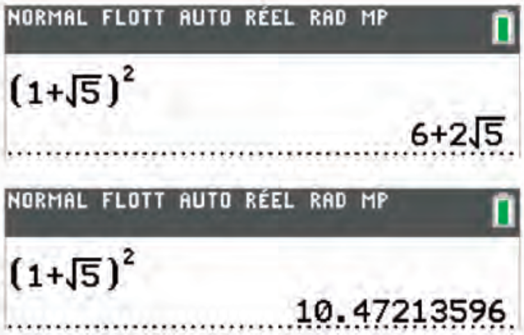



On trouve alors que $17^2 - 5^2 = 264$.

Exemple 2 : Calculer $A = \left(1+\sqrt{5}\right)^2$ puis donner une valeur approchée à $10^{-3}$ près.



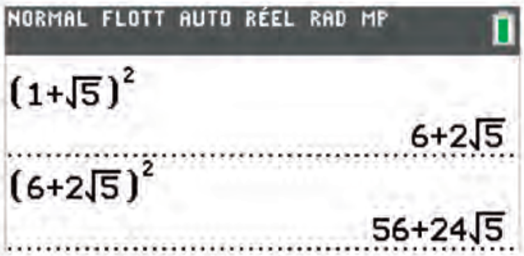




On obtient $A = 6 + 2\sqrt{5}$. Pour obtenir une valeur approchée on utilise la touche [◂▸] qui permet de passer d'une écriture exacte à une valeur approchée.

Ainsi $A \approx 10{,}472$ à $10^{-3}$ près.

Exemple 3 : En déduire la valeur exacte de $A^2$.

On va donc faire le calcul de $\left(6+\sqrt{5}\right)^2$. On commence par ouvrir une parenthèse, puis on va récupérer le résultat précédent à l'aide des flèches de direction (et on valide en appuyant sur [entrer] pour copier-coller la sélection en bleu).


On obtient ainsi $A^2 = 56 + 24\sqrt{5}$.

# Calculs de carrés, de puissances…

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Touche exposant [ ^ ]

Pour écrire des puissances comme sur son cahier (avec un exposant) on utilise la touche [ ^ ].

Exemple 1 : Développer et simplifier l'expression $(2-\sqrt{5})^3+1$.

Ce calcul est assez long à faire, il est donc intéressant de vérifier notre résultat avec sa calculatrice.

Pour élever notre nombre à la puissance 3 on utilise [ ^ ] mais n'oubliez pas de « sortir » de l'exposant à l'aide de la flèche de droite.



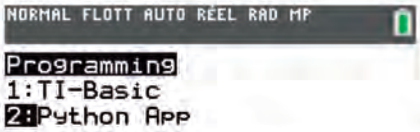
On trouve ainsi $(2-\sqrt{5})^3+1=39-17\sqrt{5}$

## Carré et puissance avec Python

Pour effectuer un calcul avec une puissance en Python, on utilise le symbole suivant :`**`.

Ainsi $5^2$ s'écrit en Python **`5**2`**.

Pour accéder à Python on appuie sur [prgm] puis on choisit **`Python App`**.





On arrive alors dans le gestionnaire de script (comme un gestionnaire de fichier). Mais ne nous en préoccupons pas maintenant, allons dans la `console` appelé aussi le `Shell` :

Exemple 2 : Calculons $5^2$ puis $17^5$ en console Python :


Vous pouvez appuyer sur [x²] pour calculer $5^2$ et sur [ ^ ] pour calculer $17^5$, ça marche !

**Remarque :** Une fois que vous avez terminé, pour sortir de Python, il faut faire [2nde] [mode] (quitter) et confirmer en appuyant sur **`Ok`**.

# Notation scientifique, sin cos…



## Notation scientifique

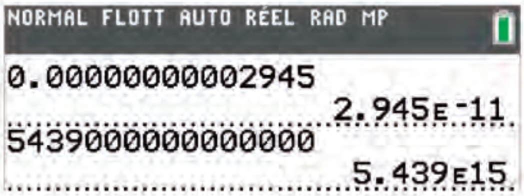
En sciences on manipule des nombres parfois très petit (chimie, biologie) et parfois très grand (physique, astronomie).

Il est pénible d'écrire un petit nombre de la forme $x = 0{,}000\,000\,000\,029\,45$ ou bien encore un grand nombre $y = 5\,439\,000\,000\,000\,000$.

Afin de travailler plus facilement avec ces nombres, on utilise la « notation scientifique ».

**Définition** : L'écriture d'un nombre sous la forme $a \times 10^n$ ou $-a \times 10^n$ avec $a \in [1; 10[$ et $n \in \mathbb{Z}$ est appelée écriture scientifique.

Exemple 1 : L'écriture scientifique du nombre $x$ précédent est $2{,}945 \times 10^{-11}$ de même l'écriture scientifique de $y$ est $5{,}439 \times 10^{15}$.



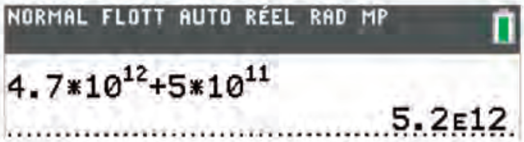

Notre calculatrice utilise naturellement l'écriture scientifique, mais elle ne l'écrit pas exactement comme nous (voir écran ci-contre).

Rappelons au passage qu'en France on écrit les nombres décimaux à l'aide d'une virgule mais les anglo-saxons utilisent [ . ].

D'autre part E signifie « 10 puissance ».

Ainsi **5.439E15** s'écrira avec nos notations habituelles $5{,}439 \times 10^{15}$.

Exemple 2 : Donner l'écriture scientifique de $z = 4{,}7 \times 10^{12} + 5 \times 10^{11}$.




On entre notre expression dans notre calculatrice. On peut utiliser [ ^ ] pour entrer les exposants des puissances de 10.

La calculatrice comprend notre écriture avec des puissances de 10 mais elle répond avec son fameux E (exposant) ce qui donne avec nos notations usuelles : $z = 5{,}2 \times 10^{12}$.

**Remarques** : La calculatrice utilise l'écriture scientifique lorsqu'elle ne peut pas afficher tous les chiffres composant notre nombre à l'écran.

Il est possible de forcer l'écriture scientifique en changeant de [mode] : Mais voici le paramétrage que nous vous conseillons :

**NORMAL** est l'affichage classique :
Le nombre 53 est affiché **53**.
**SCI** utilise systématiquement l'écriture scientifique donc 53 est affiché **5.3E1**.

**MATHPRINT** permet d'afficher $\frac{5}{3}$ plutôt que 5/3.


**FLOTTANT** adapte l'affichage des décimales.
**3** permet d'arrondir systématiquement avec 3 chiffres après la vigule.

**DEGRE** est l'unité utilisée pour les angles en classe de seconde.

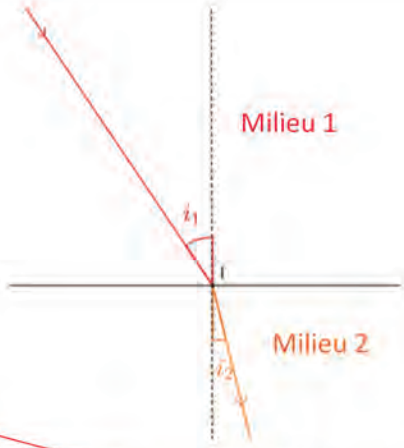
# Notation scientifique, sin cos…

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## sinus et cosinus

Exemple 1 : Loi de Snell-Descartes

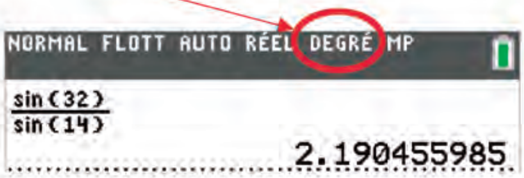



On cherche à calculer $n_2$, l'indice de réfraction du milieu 2, connaissant : $n_1 = 1$ l'indice de réfraction du milieu 1, $i_1 = 32°$ l'angle d'incidence et $i_2 = 14°$ l'angle de réfraction.

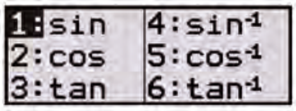
On sait d'après la loi de Snell-Descartes que $n_1 \sin(i_1) = n_2 \sin(i_2)$

Ainsi $n_2 = \frac{n_1 \sin(i_1)}{\sin(i_2)} = \frac{1 \times \sin(32°)}{\sin(14°)}$.

Prenons notre calculatrice et vérifions bien que l'unité des angles est le degré, puis effectuons notre calcul :




La fonction **sin** est accessible dans [trig] .

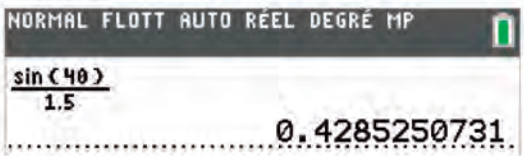



Si votre calculatrice n'est pas en mode degré, pas de panique !

Appuyer sur [mode] et sélectionner **DEGRÉ**.


Exemple 2 : On donne maintenant $n_1 = 1$ ; $n_2 = 1{,}5$ ; $i_1 = 40°$ et on demande de trouver l'angle $i_2$.

On sait que $n_1 \sin(i_1) = n_2 \sin(i_2)$ donc $\sin(i_2) = \frac{n_1 \sin(i_1)}{n_2} = \frac{1 \times \sin(40°)}{1{,}5}$




On trouve ainsi $\sin(i_2) \approx 0{,}429$.

Maintenant pour trouver $i_2$, il faut utiliser **sin**⁻¹ accessible dans [trig] :


On obtient ainsi $i_2 \approx 25{,}4°$.

# Représentation graphique de fonctions

TI-82 Advanced Edition Python

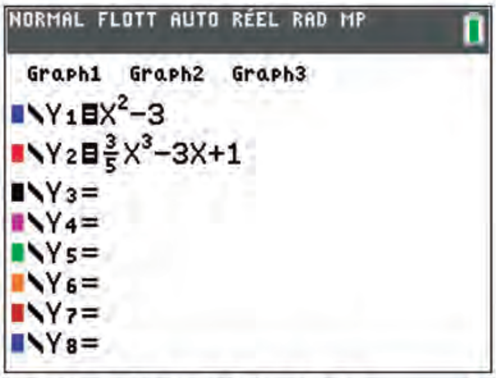
TI-83 Premium CE Edition Python

## Contexte

La calculatrice trouve pleinement sa place dans l'étude des fonctions, en seconde. L'observation de fonctions de références ou la découverte de fonctions nouvelles nécessite de maîtriser notamment les techniques de représentation, en ajustant les fenêtres de représentations, les couleurs, les styles de tracés et bien d'autres paramètres que nous allons découvrir.

On souhaite représenter les fonctions définies sur $\mathbb{R}$, $f : x \to x^2 - 3$ et $g : x \to \frac{3}{5}x^3 - 3x + 1$

## 1. Saisir l'expression de la fonction

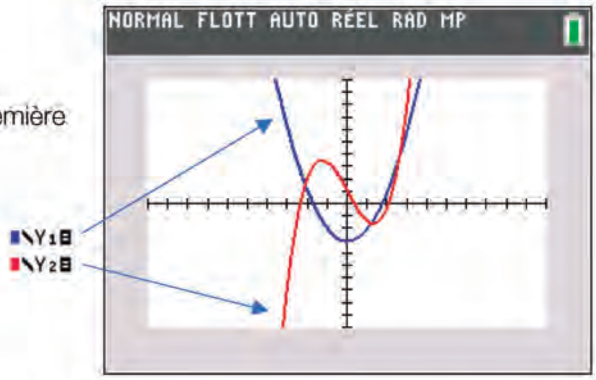



Pour commencer la saisie, on appuie sur la touche f(x)

On peut alors saisir les expressions souhaitées.

On utilise la touche X,T,θ,n pour obtenir la variable $x$ dans la saisie.

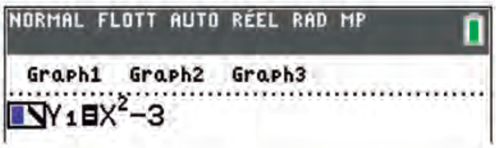
On n'oublie pas non plus [fraction], $x^2$ et ^ pour la saisie de coefficients fractionnaires, élever au carré ou plus généralement à tout autre exposant.




On appuie sur la touche graphe pour obtenir immédiatement une première représentation graphique.

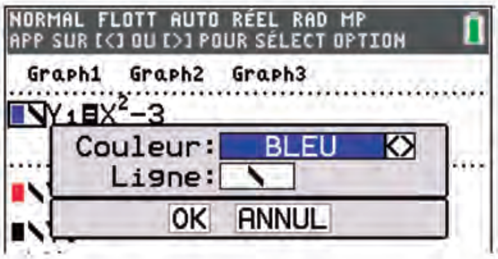
La couleur de la courbe représentative est définie dans l'éditeur de fonctions, ici, par défaut, bleu pour Y1 et rouge pour Y2.

## 2. Modifier la représentation graphique

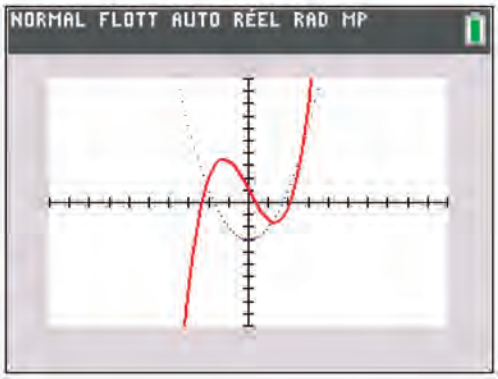



On retourne dans l'éditeur de fonctions ( f(x) ).

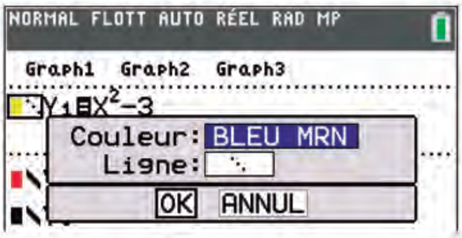
A l'aide des flèches de la calculatrice ( ‹ ou › ), on déplace le curseur pour le positionner sur l'indicateur de couleur. Celui est désormais mis en valeur par un cadre clignotant.

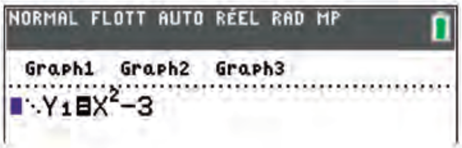



On valide avec la touche entrer pour accéder au menu de configuration.

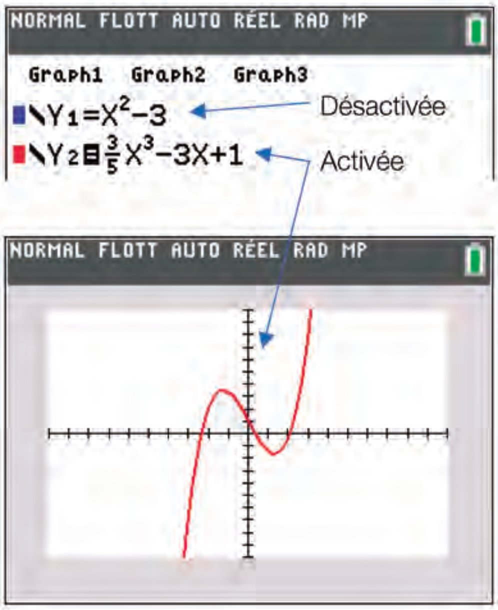



On modifie les paramètres de représentations comme la couleur mais aussi la forme de représentation (trait épais, fin, pointillé…)




Enfin, il est possible de désactiver temporairement l'affichage d'une courbe représentative en positionnant le curseur sur le signe égal de la fonction que l'on ne souhaite plus « affichée ». On appuie sur la touche [entrer] pour activer/réactiver le dessin. Le signe égal apparait sur fond noir (dessin activé) ou sur fond blanc (dessin désactivé)




## 3. Modifier le repère avec le menu ZOOM

Par défaut, le repère est configuré pour représenter les abscisses sur l'intervalle [-10 ;10].

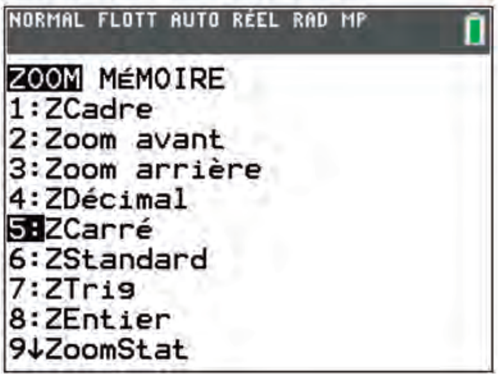
De même pour les ordonnées.

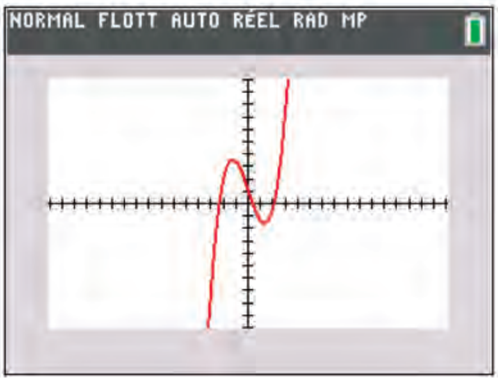
Cependant, l'écran de la calculatrice est rectangulaire donc par défaut le repère n'est pas orthonormé. C'est-à-dire que les graduations « horizontales » et « verticales » sont différentes.

On appuie sur la touche [zoom] pour accéder aux différentes options.

Par exemple, pour rendre automatiquement le repère orthonormé, on choisit `ZCarré`.

L'option `ZStandard` permet de reconfigurer la fenêtre par défaut.

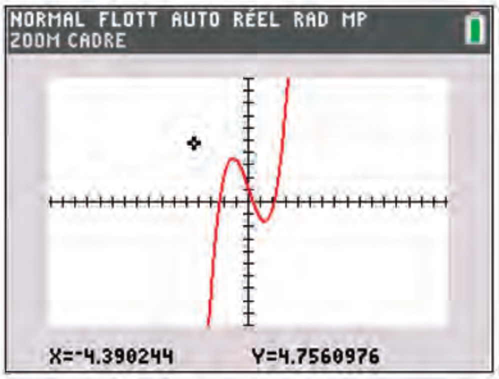


# Représentation graphique de fonctions

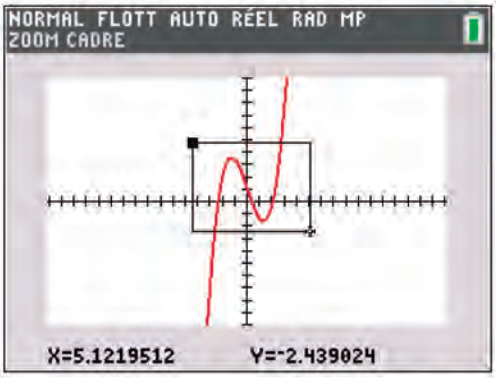
TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python




Une fois notre représentation obtenue, on souhaite mieux observer une zone précise de la courbe comme celle du changement de variations autour de l'intersection des axes. Pour cela, on a un outil, c'est l'option **ZCadre**.

Lorsqu'elle est sélectionnée, le curseur change de forme et prend la forme d'une croix.

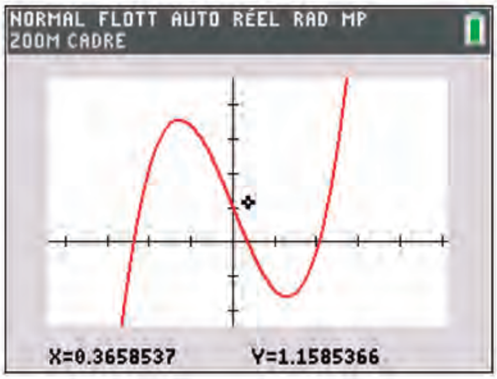



On va définir une zone rectangulaire qui va représenter la zone sur laquelle on veut recentrer la représentation.

Pour cela, on déplace le curseur, à l'aide des flèches de la calculatrice, dans un premier coin de notre rectangle et on valide avec la touche entrer .

On déplace ensuite le curseur vers le deuxième coin et on valide à nouveau avec la touche entrer .

Durant le déplacement, on visualise le rectangle qui se dessine au fur et à mesure ainsi que les coordonnées du curseur.



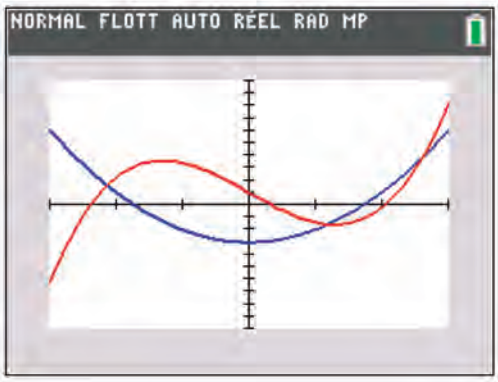

Pour sortir du mode **ZCadre**, on appuie sur la touche annul .

## 4. Modifier le repère avec le menu Fenêtre

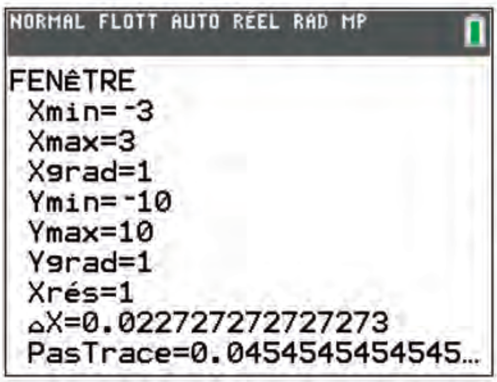
Il est également possible d'agir directement sur notre fenêtre de représentation en appuyant sur la touche fenêtre.

Par exemple, lorsqu'on dispose d'un intervalle de définition pour les fonctions à analyser.

Réactivons l'affichage de **Y1** et observons la représentation sur l'intervalle [-3 ; 3].




Pour cela, dans le menu fenêtre ( fenêtre ), on saisit les valeurs -3 pour **Xmin** et 3 pour **Xmax**.

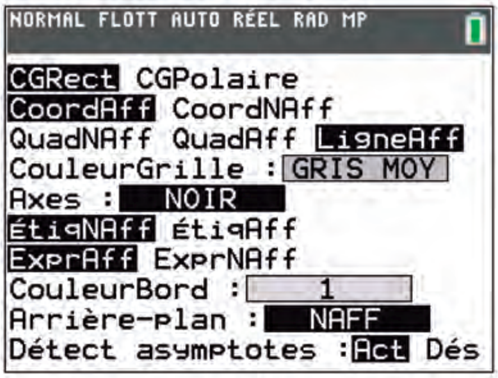



On n'oublie pas de valider avec la touche entrer

On retourne alors dans la représentation graphique graphe

Celle-ci est bien représentée sur [-3 ;3]

## 5. Faciliter la lecture graphique



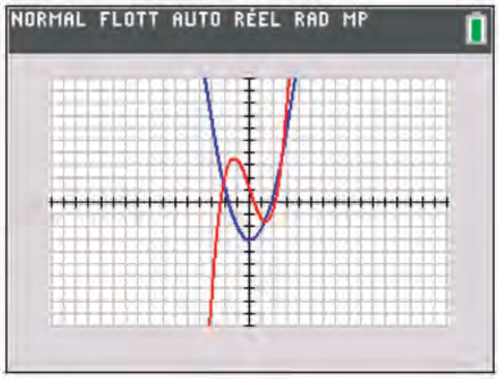

La calculatrice permet d'afficher un quadrillage pour faciliter la lecture graphique.

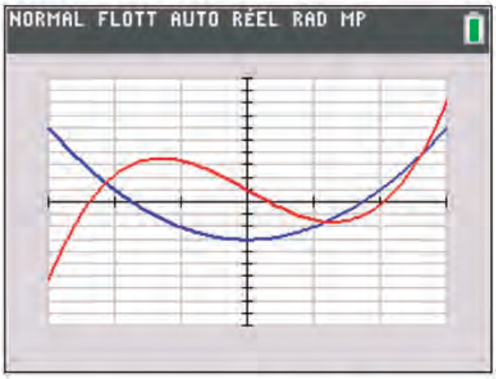
Pour l'activer, il faut se rendre dans le menu **format**, accessible via la combinaison de touche [2nde] + [zoom].

Il faut alors valider l'option **LigneAff** en déplaçant le curseur dessus et en appuyant sur [entrer].

Lorsqu'on retourne dans la fenêtre graphique, on constate la présence du quadrillage en fond gris.

Pour le désactiver, il faut sélectionner dans le menu **format**, l'option **QuadNAff**



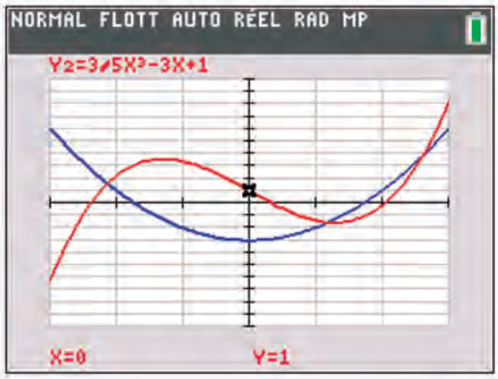

Il existe une autre façon de faciliter la lecture graphique. Il s'agit [trace] iver le mode **trace** à l'aide de la touche

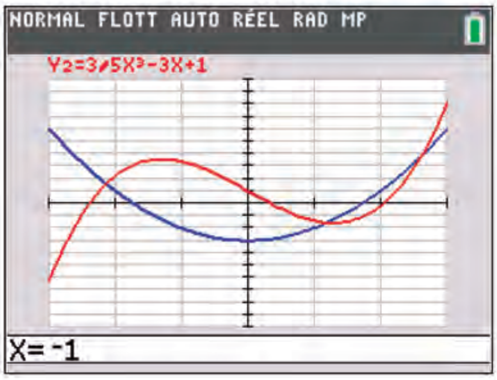
Dans ce cas, on déplace le curseur à l'aide des flèches [<] ou [>]. Le curseur reste sur la courbe représentative de la fonction sélectionnée et les coordonnées du point (antécédent ; image) s'affichent.

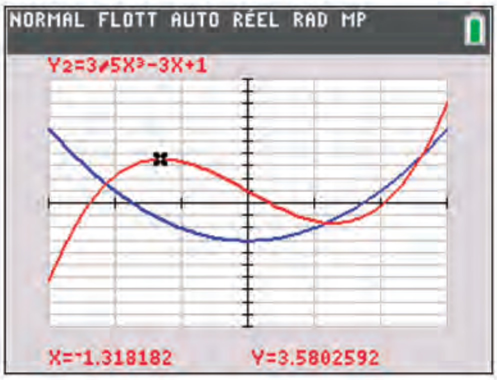
Il est également possible de saisir directement la valeur souhaitée pour $x$.

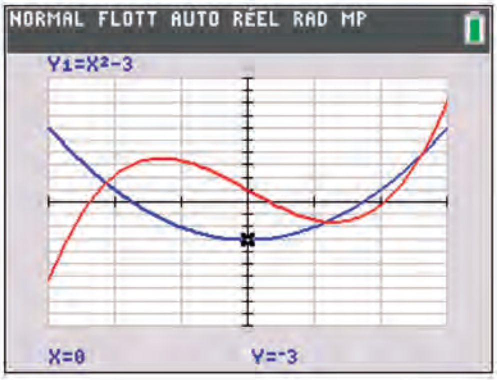
On peut passer d'une courbe représentative à l'autre à l'aide des flèches [v] ou [^]. L'affichage prend alors la couleur de la courbe représentative parcourue.

On quitte le mode **trace** à l'aide de la touche [annul]



# Fonctions, images, antécédents et tableau de valeurs

TI-82 Advanced Edition Python

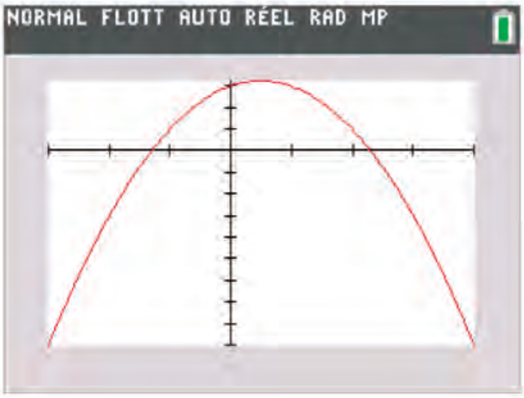
TI-83 Premium CE Edition Python

## Images

Soit $f$ la fonction définie sur $[-3; 4]$ par $f(x) = -x^2 + x + 3$.

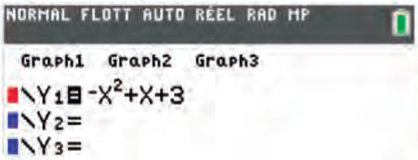
1. Déterminer graphiquement les images de $-2$ et $\sqrt{2}$ par la fonction $f$.
2. Déterminer par le calcul les images de $-2$ et $\sqrt{2}$ par la fonction $f$

## 1. Méthode graphique




On commence par représenter graphiquement la fonction.

On entre l'expression de la fonction en appuyant sur [f(x)], puis on entre l'ensemble de définition en appuyant sur le bouton suivant [fenêtre] :

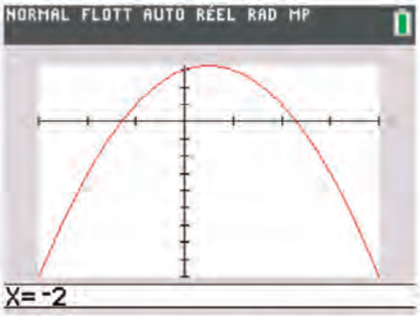


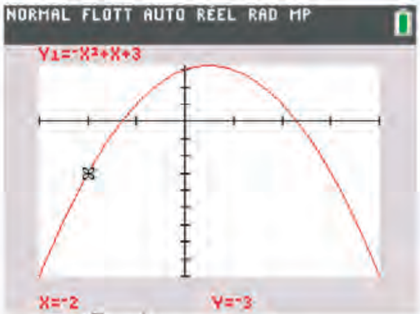


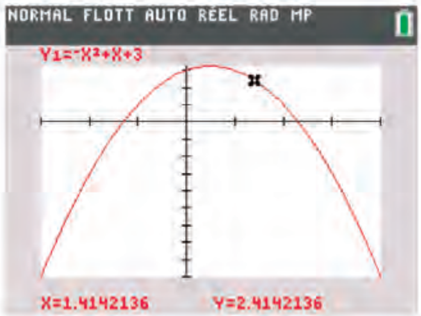
Pour ajuster la fenêtre automatiquement on appuie sur [zoom] `0:AjustZoom`

Maintenant à l'aide de l'outil graphique [2nde] [trace] (calculs f4) `Image` on entre $-2$

Attention ! Il faut utiliser la touche [(-)] (c'est le signe moins de l'opposé).



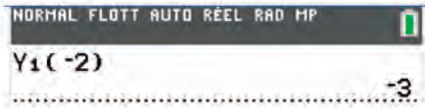

On entre maintenant $\sqrt{2}$ en appuyant sur [2nde] [x²] puis [entrer].

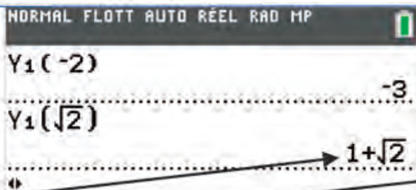
Ainsi l'image de $-2$ par la fonction $f$ est $-3$ et l'image de $\sqrt{2}$ est <u>environ</u> $2{,}41$.

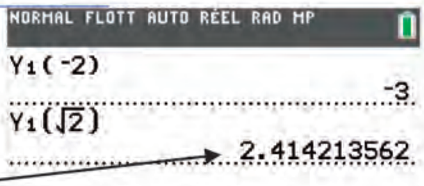
Cette méthode ne permet d'obtenir, comme toute méthode graphique, que des valeurs approchées.

## 2. Méthode par le calcul

Quitter la page graphique (appuyer sur [2nde] [mode] (quitter)). Pour calculer $f(-2)$, entrer `Y1(-2)` en appuyant sur [var] onglet **VAR Y** puis choisir `Fonction…`




On obtient ici la valeur exacte, puis une valeur approchée si besoin (touche [◂▸]).

# Fonctions, images, antécédents et tableau de valeurs

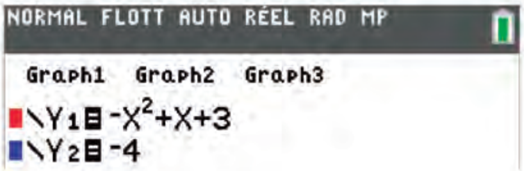
TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

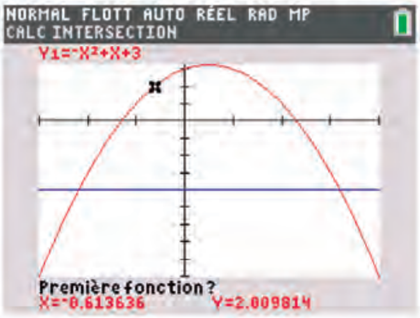
## Antécédents

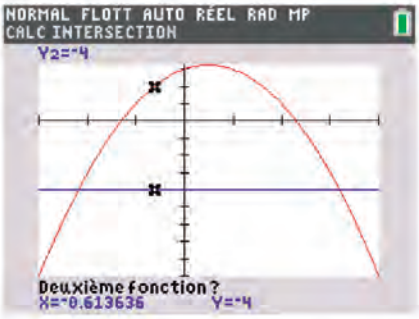
En reprenant la fonction précédente, déterminer graphiquement les éventuels antécédents négatifs de $-4$.

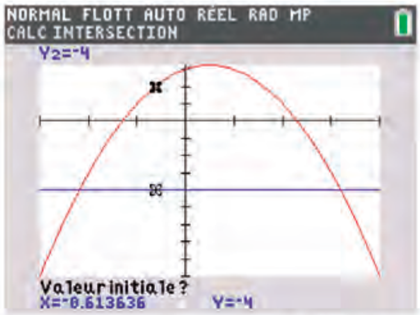
Pour cela on va afficher la droite d'équation $y=-4$ en utilisant la fonction `Y2` en appuyant sur fenêtre.



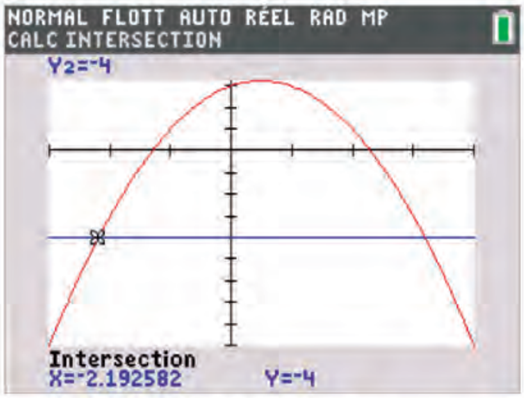

Puis dans [2nde] [trace] (calculs f4) on sélectionne `intersection`.




Il faut choisir une valeur initiale proche du point d'intersection recherché.




Conclusion : L'antécédent négatif de $-4$ par $f$ est environ $-2{,}19$. Ici on ne peut obtenir qu'une valeur approchée car on a fait une lecture graphique.

## Tableau de valeurs

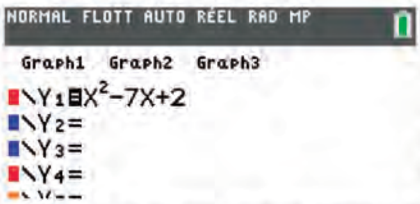
<u>Exemple 3</u> : Soit $f$ la fonction définie sur $[-1;2]$ par $f(x)=x^2-7x+2$.

Compléter le tableau de valeurs de $f$ ci-dessous :

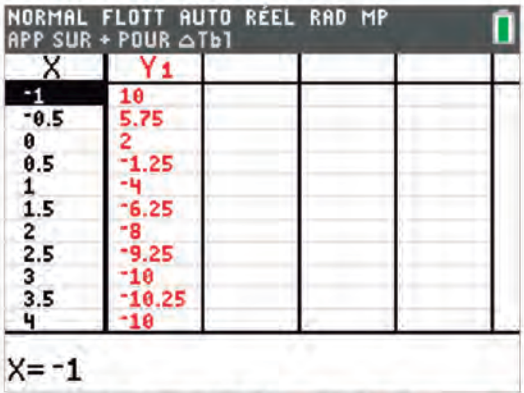
| $x$ | $-1$ | $-0{,}5$ | $0$ | $0{,}5$ | $1$ | $1{,}5$ | $2$ |
|---|---|---|---|---|---|---|---|
| $f(x)$ | | | | | | | |

On commence par entrer l'expression de la fonction en appuyant sur [f(x)].

Avant d'afficher le tableau de valeurs, il faut le paramétrer. Pour cela appuyer sur [2nde] [fenêtre] (déf table f2).



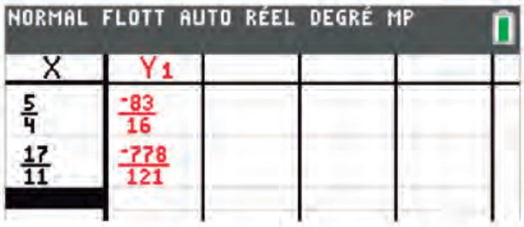





Pour obtenir maintenant le tableau de valeurs on appuie sur [2nde] [graphe] (table f5).

**Remarque** : Il est possible d'obtenir des images de valeurs qu'on souhaite (en les entrant une à une) en sélectionnant `Indpnt : Demande`.

```
Indpnt :  Auto Demande
Dépndte : Auto Demande
```



# Fonctions – Maximum et minimum

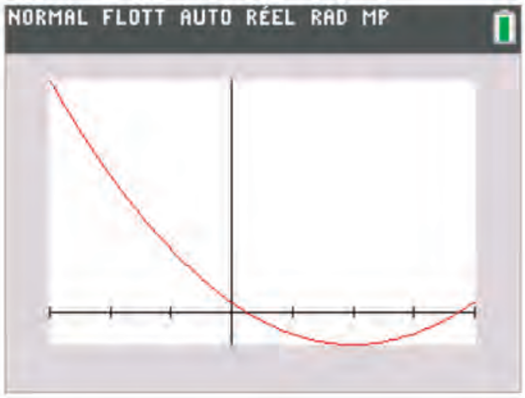
TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

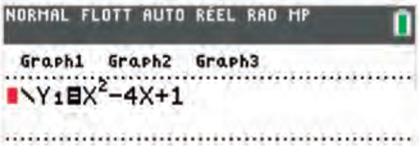
## Recherche graphique d'extrema

Soit $f$ la fonction définie sur $[-3; 4]$ par $f(x) = x^2 - 4x + 1$.

Représenter graphiquement cette fonction sur votre calculatrice puis déterminer son minimum.



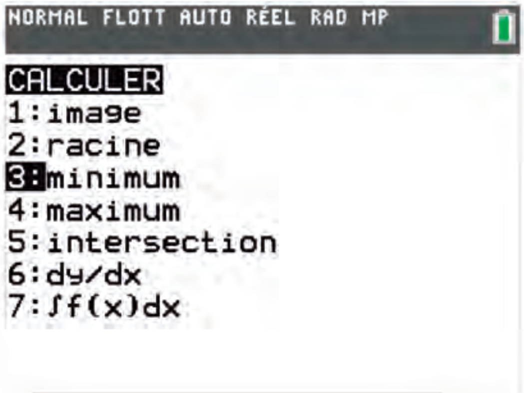

Pour représenter graphiquement $f$ on entre son expression en appuyant sur [f(x)] puis on paramètre la fenêtre (touche [fenêtre]) en entrant **Xmin=-3** et **Xmax=4** (ce qui correspond à l'ensemble de définition de $f$).



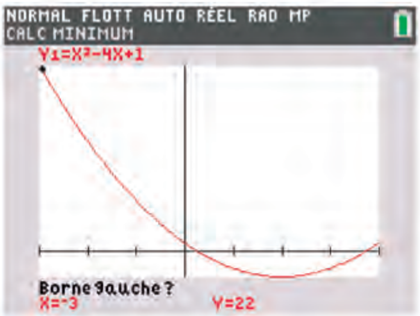


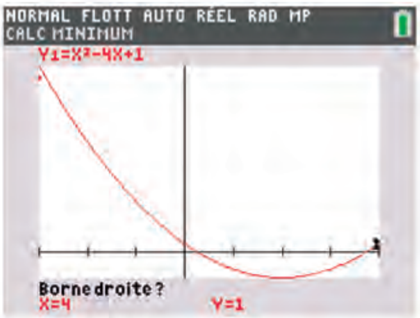
On effectue un ajustement automatique de la fenêtre en appuyant sur [zoom] puis sur **0:AjustZoom**.

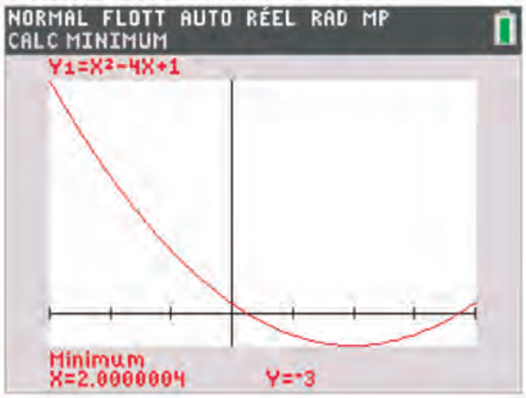
Pour trouver le minimum de cette fonction on appuie sur [2nde] [trace] (calculs f4) et on choisit **minimum**.




1. Sélectionner la borne gauche et appuyer sur [entrer]. On peut déplacer le curseur à l'aide des touches de direction ou bien en entrant directement la valeur de la borne inférieure au clavier.
2. Sélectionner la borne gauche et appuyer sur [entrer].
3. Appuyer une nouvelle fois sur [entrer] pour la valeur initiale.




Conclusion : Le minimum de $f$ sur $[-3; 4]$ est $-3$ et il est atteint en $x = 2$.

Ces valeurs obtenues restent des valeurs approximatives car elles sont issues d'une lecture et d'une recherche graphique.

# Fonctions – Maximum et minimum

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Recherche d'extrema avec Python

On donne le script Python suivant de la fonction **minimum**. Cette fonction prend comme argument une fonction mathématique notée **g** (dont on cherche le **minimum**), et renvoie le **minimum** de **g** sur l'intervalle [**-3,4**] en utilisant la méthode par balayage.

1. Recopier ce script et définir en Python la fonction $f$ de l'exemple précédent.

2. Exécuter ce script et déterminer le minimum de $f$ sur $[-3;4]$.

3. Ecrire de la même façon la fonction Python **maximum**.

```
ÉDITEUR : FONCTION
LIGNE DU SCRIPT 0011
def minimum(g):
  n=1000
  min=g(-3)
  for i in range(n+1):
    x=-3+i*7/n
    if g(x)<min:
      min=g(x)
  return min
Fns… | a A # | Outils | Exéc | Script
```

## 1. Fonction Python f

On recopie le script de la fonction maximum puis on écrit celui de la fonction **f** de la partie précédente (voir ci-contre).

```
ÉDITEUR : FONCTION
LIGNE DU SCRIPT 0019
def f(x):
  y=x**2-4*x+1
  return y
```

## 2. Déterminer le minimum de f

Pour exécuter le script on appuie sur **Exéc**. Dans la console on appelle notre fonction en appuyant sur [var] puis on entre les paramètres (voir copie d'écran ci-contre).

On retrouve bien que le minimum de $f$ sur $[-3;4]$ est environ $-3$. Cet algorithme ne donne qu'une valeur approchée du minimum.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de FONCTION
>>> from FONCTION import *
>>> minimum(f)
-2.999996
>>> |
Fns… | a A # | Outils | Éditer | Script
```

## 3. Script fonction maximum

La fonction maximum est très similaire à la fonction minimum. La seule ligne qui change est **if g(x)>max** (à la place de **if g(x)<min**).

```
ÉDITEUR : FONCTION
LIGNE DU SCRIPT 0027
def maximum(g):
  n=1000
  max=g(-3)
  for i in range(n+1):
    x=-3+i*7/n
    if g(x)>max:
      max=g(x)
  return max
Fns… | a A # | Outils | Exéc | Script
```

# Résolution graphique d'équations

# $f(x) = k$

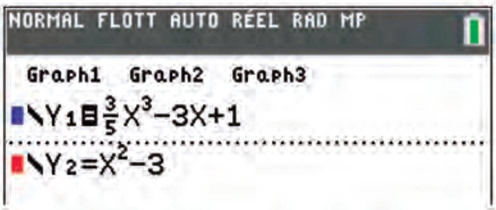
TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## Contexte

On va utiliser la calculatrice pour résoudre, dans $\mathbb{R}$, graphiquement l'équation $\frac{3}{5}x^3 - 3\,x + 1 = 2$

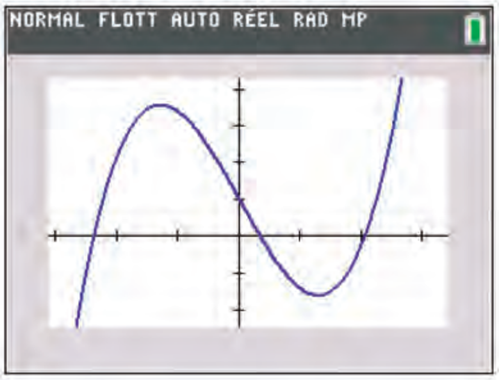
## 1. Préparer sa représentation graphique




On commence par saisir dans **Y1** le membre de gauche de notre équation, elle contient l'expression $\frac{3}{5}x^3 - 3\,x + 1$

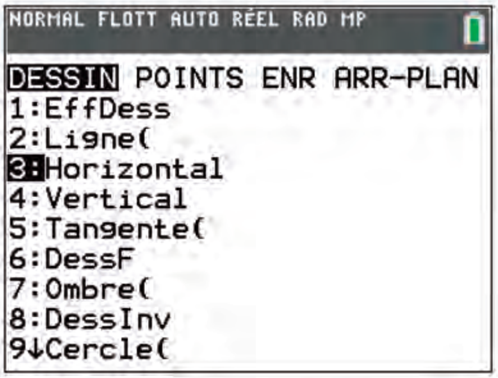
On s'assure que seule la fonction qui nous intéresse est activée ( ).

On vérifie visuellement qu'aucune représentation statistique n'est activée (**Graph1**, **Graph2** et **Graph3** doivent être sur fond blanc)




On applique les techniques de représentations de fonctions vu précédemment pour sélectionner la zone qui nous intéresse. Dans cet exemple, on travaille autour de l'ordonnée 2 (membre de droite notre équation).

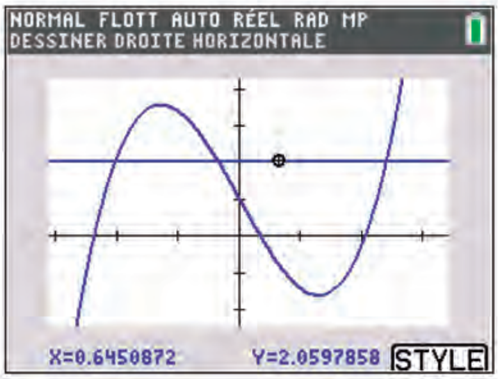
## 2. Dessiner une droite horizontale




On peut commencer par une première manipulation très rapide et qui a l'avantage de se réaliser directement à partir de la représentation graphique.

A l'aide de la combinaison de touches 2nde + prgm , on se rend dans le menu **dessin** et on sélectionne **Horizontal.**

On a alors la possibilité de pouvoir positionner, à l'aide des flèches + de la calculatrice, une droite horizontale directement sur notre représentation graphique.




On obtient un premier aperçu des solutions de notre équation en positionnant la droite à l'ordonnée 2.

On recherche donc 3 solutions pour notre équation.

On peut effacer nos tracés à l'aide de la fonction **EffDess**.

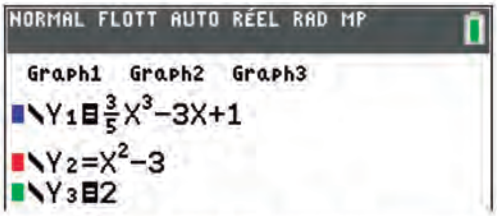
Cette méthode permet un aperçu rapide mais pour obtenir des approximations numériques de nos solutions, nous allons opter pour autre méthode.

# Résolution graphique d'équations

# $f(x) = k$



## 3. Représenter une fonction constante



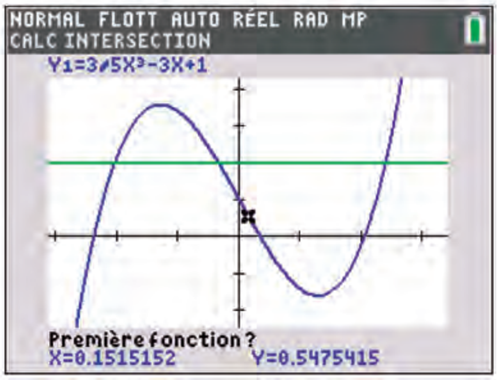

On souhaite obtenir une approximation numérique des solutions de notre équation, on va donc créer une deuxième représentation graphique de la forme $y = k$ pour pouvoir exploiter les fonctionnalités de calculs offertes.

Concrètement, dans notre exemple, on va représenter dans **Y3** la droite d'équation $y = 2$. On modifie, en vert, sa couleur de représentation.

**Y2** est désactivée mais conservée.

On se rend dans le menu **calculs** à l'aide des touches 2nde + trace

On sélectionne l'item **5:intersection**



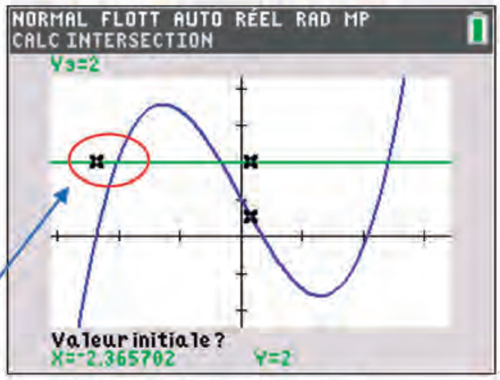

On est automatiquement redirigé vers la représentation graphique.

On sélectionne la première fonction **Y1**, l'écriture apparait en bleu et le curseur est positionné sur la courbe représentative. On valide avec la touche entrer

On recommence l'opération pour sélectionner la deuxième fonction **Y3** en vert.

Enfin un troisième curseur apparait à l'écran et nous invite à définir la valeur initiale. Attardons-nous un instant sur cette partie.

La calculatrice nous demande, en fait, de prédéfinir la zone de recherche de la solution souhaitée.



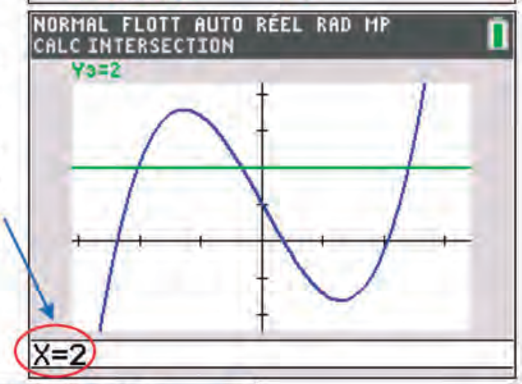

En effet, on a vu qu'il y avait, dans notre exemple, 3 solutions possibles.

On va commencer par l'approximation de la solution d'abscisse la plus petite (donc la plus à gauche de l'écran).

Pour cela on déplace le curseur à l'aide des flèches de la calculatrice ‹ ›

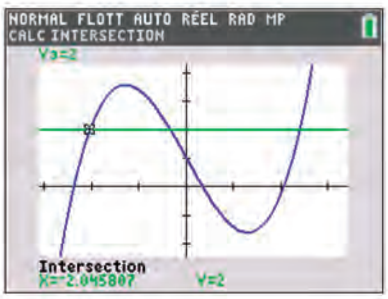
Une fois positionné à proximité de l'intersection voulue, on valide entrer

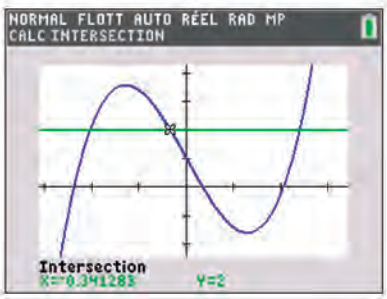
La calculatrice positionne le curseur sur l'intersection repérée et fournie une approximation de la valeur de $x$, ici $-2{,}045807$

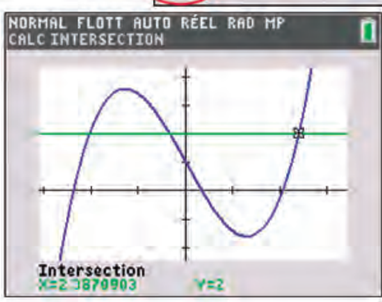



On recommence les mêmes manipulations pour les recherches des deux autres solutions. On peut saisir directement sur le pavé numérique les abscisses voulues. Cela active automatiquement une zone de saisie en dessous de la fenêtre et permet de gagner du temps en économisant sur le déplacement du curseur. On valide avec la touche entrer

Finalement on obtient solutions $x \approx -2{,}05$ *ou* $x \approx -0{,}34$ *ou* $x \approx 2{,}39$



# Vérification d'un tableau de signes

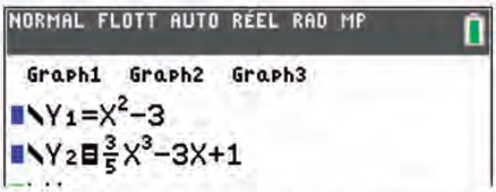
TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

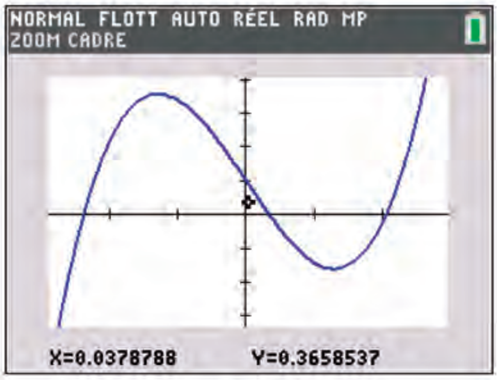
## Contexte

On va utiliser les fonctionnalités de la calculatrice pour vérifier à l'aide de la représentation graphique, le tableau de signes de la fonction définie sur $\mathbb{R}$, $g : x \to \frac{3}{5}x^3 - 3\,x + 1$

## 1. Préparer sa représentation graphique



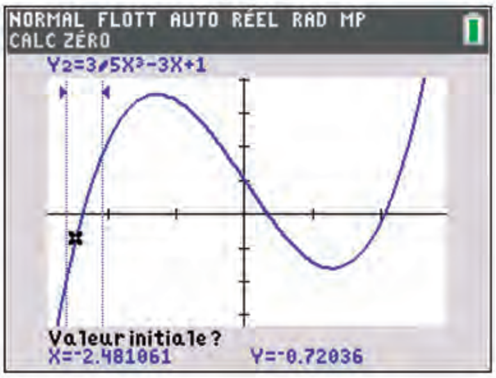

On configure notre éditeur de fonctions en ne laissant activé que la représentation de la fonction $g$, dont l'expression a été saisie dans **Y2**. Elle sera représentée en bleu. On souhaite conserver l'expression de **Y1** pour d'autres manipulations éventuelles.




A l'aide de la fonction **ZCadre** du menu **ZOOM** [zoom] ,on obtient la représentation ci-contre.

L'utilisation de cette fonction a déjà été décrite dans la fiche « 4 - Représentation graphique de fonctions »

## 2. Rechercher les racines de la fonction




Il s'agit, dans le cadre de la vérification du tableau de signes de $g$, de vérifier les valeurs approchées de ses trois racines.

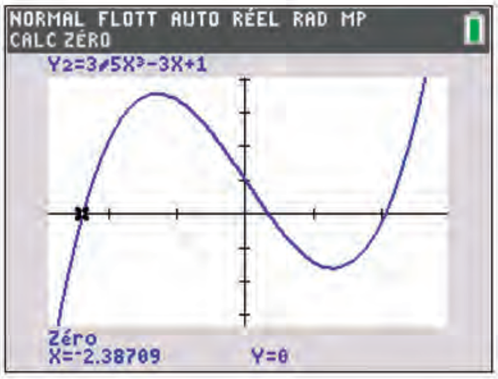
On utilise pour cela la fonction **racine** du menu **calculs** ( [2nde] + [trace] )

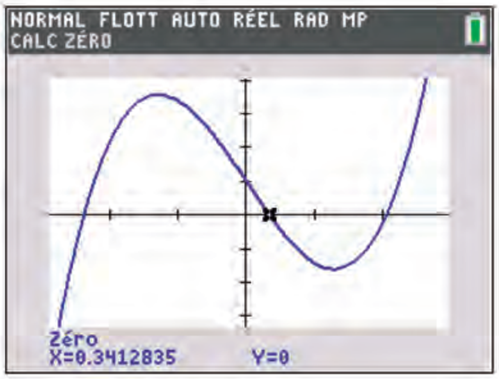
Son fonctionnement est similaire aux fonctions **intersection** et **minimum** décrites dans les fiches 6 et 7 précédentes.

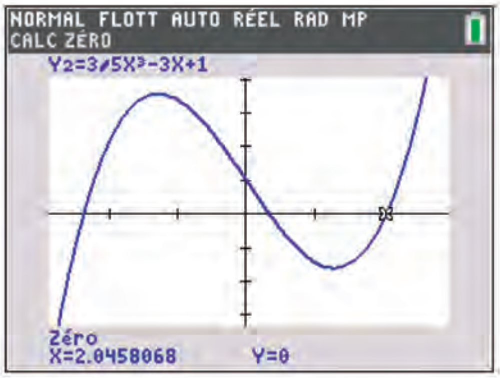
Borne gauche, borne droite, valeur initiale sont définies et validées.

L'opération est réalisée trois fois pour chacune des racines observables sur la représentation graphique.

Nous obtenons les valeurs approchées $x \approx -2{,}39 \; ou \; x \approx 0{,}34 \; ou \; x \approx 2{,}05$



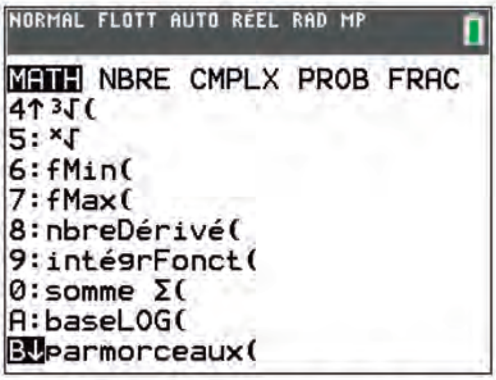



# Vérification d'un tableau de signes



## 3. Mettre en évidence graphiquement le signe de la fonction

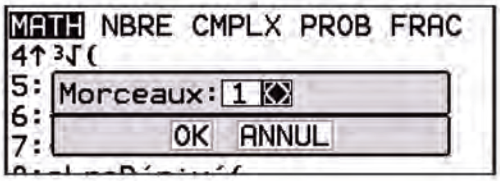
On veut dessiner, en vert, les morceaux de la courbe représentative de $g$ où la fonction est positive et en rouge, les morceaux où elle est négative.




Pour cela, on positionne le curseur dans l'éditeur de fonctions sur la ligne $Y_3$.

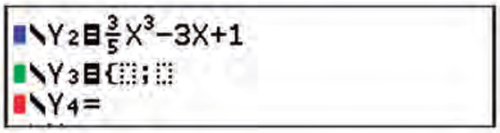
La commande dont nous allons avoir besoin est accessible dans le menu **MATH** de la calculatrice via la touche math

Il s'agit de l'item `B:parmorceaux(`. Il faut utiliser les flèches de la calculatrice pour faire défiler les différentes instructions jusqu'à l'item B.



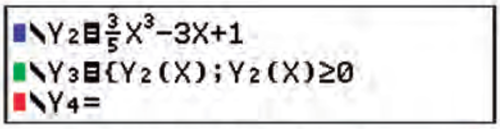

Cette commande permet la représentation conditionnelle de fonctions, c'est-à-dire qu'elle ne tracera la fonction désirée que si la condition mentionnée est réalisée.

On sélectionne `Morceaux : 1`.




On va demander à tracer $Y_2$ en vert, lorsque $Y_2$ est positive.

Pour cela, on modifie la couleur de tracé par défaut de $Y_3$. On choisit le vert.

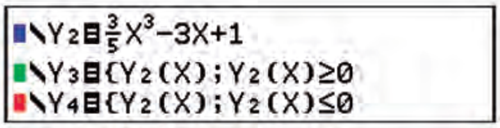



On complète la saisie de la manière ci-contre. Le nom de variable $Y_2$ ne doit pas être écrit en toutes lettres mais doit être recherché dans le menu **VAR Y** de la calculatrice via la touche var . On accède ensuite au menu **Fonction...** pour sélectionner $Y_2$.

Le signe ≥ s'obtient dans le menu **tests** via les touches 2nde + math

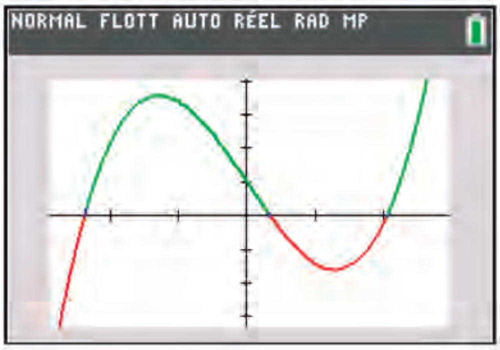


Une fois cette saisie réalisée, on se place dans $Y_4$, configuré pour tracer en rouge et l'on saisit, à nouveau à l'aide de la commande `parmorceaux`, la définition ci-contre, qui demande de tracer $Y_2$ en rouge lorsque $Y_2$ est négative.




On peut vérifier graphiquement le tableau de signes de la fonction $g$.

| $x$ | $-\infty$ | | $-2{,}39$ | | $0{,}34$ | | $2{,}05$ | | $+\infty$ |
|---|---|---|---|---|---|---|---|---|---|
| $g(x)$ | | $-$ | 0 | $+$ | 0 | $-$ | 0 | $+$ | |



# Résolution graphique d'inéquations $f(x) \leq g(x)$

TI-82 Advanced Edition Python

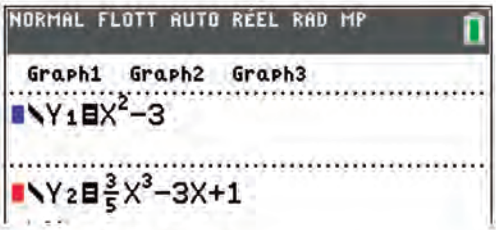
TI-83 Premium CE Edition Python

## Contexte

On va utiliser la calculatrice pour résoudre, dans $\mathbb{R}$, graphiquement l'inéquation $x^2 - 3 \leq \frac{3}{5}x^3 - 3\,x + 1$

On va, pour cela, représenter les fonctions définies sur $\mathbb{R}$, $f : x \to x^2 - 3$ et $g : x \to \frac{3}{5}x^3 - 3\,x + 1$

## 1. Préparer sa représentation graphique

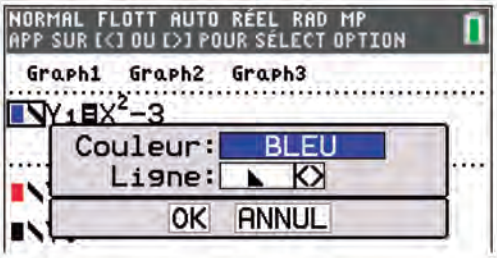



Comme on l'a déjà appris précédemment, on saisit dans **Y1** l'expression algébrique de $f$ et dans **Y2** celle de $g$.

On va maintenant agir sur le mode de représentation des courbes représentatives de chacune des fonctions

Il faut bien observer l'expression de l'inéquation que l'on souhaite résoudre :

$$\begin{array}{ccc} x^2 - 3 & \leq & \frac{3}{5}x^3 - 3\,x + 1 \\ f(x) & \leq & g(x) \\ Y_1 & \leq & Y_2 \end{array}$$

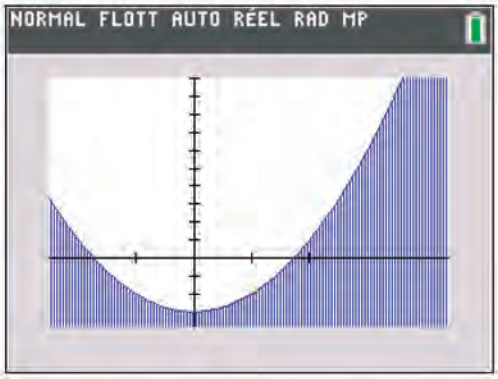



On va demander à la calculatrice de hachurer tout ce qui est « en dessous » de la courbe représentative de $f$.

Cela a donc pour effet de laisser en blanc tout ce qui est « au-dessus ».

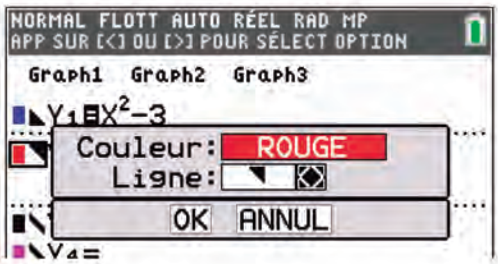
Pour cela, on positionne le curseur sur et on valide entrer

On choisit le motif pour le type de **Ligne**.




On observe dans la fenêtre « recadrée » de représentation graphique le résultat ci-contre ( graphe ).

Mathématiquement, pour une abscisse $x$ donnée, toutes les ordonnées inférieures à $f(x)$ sont donc hachurées en bleu et donc toutes les ordonnées supérieures à $f(x)$ sont laissées en blanc.




On fait un travail similaire avec la courbe représentative de $g$.

Cette fois-ci on sélectionne le type de représentation qui va hachurer tout ce qui est « au-dessus » de la courbe représentative.

Mathématiquement, pour une abscisse $x$ donnée, toutes les ordonnées supérieures à $g(x)$ sont donc hachurées en rouge et donc toutes les ordonnées inférieures à $g(x)$ sont laissées en blanc.

Thème: Analyse

# Résolution graphique d'inéquations $f(x) \leq g(x)$

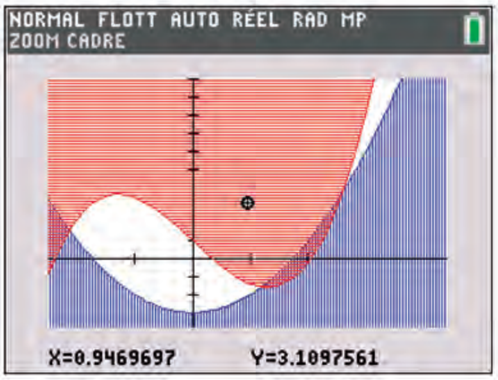
TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## 2. Interpréter le résultat obtenu

Avec cette configuration graphique, la zone répondant à notre inéquation est laissée sur fond blanc.

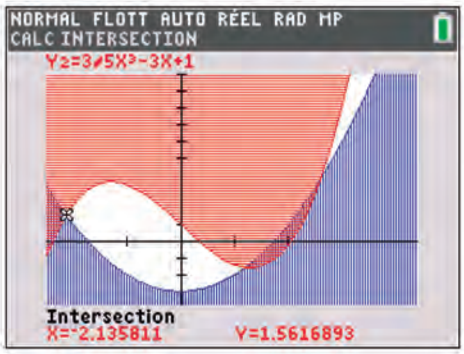
En effet, pour une abscisse $x$ donnée, il s'agit des ordonnées à la fois supérieures à $f(x)$ et inférieures à $g(x)$.

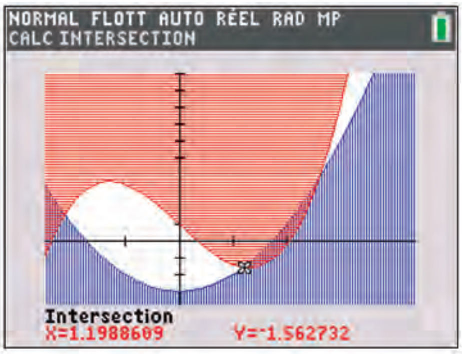


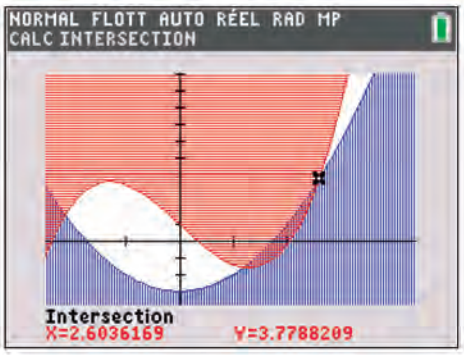

Grâce aux outils de résolution déjà rencontrés, on peut rechercher les valeurs des abscisses des points d'intersection des fonctions $f$ et $g$ pour définir l'ensemble de solutions de notre inéquation.

Ainsi à l'aide de la commande **`intersection`** disponible dans le menu **`calculs`**, on obtient l'ensemble de solutions approchées suivant.

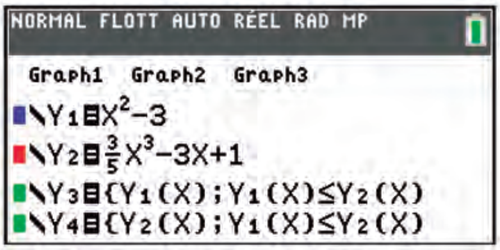
Il s'agit de l'intervalle : $[-2{,}14; 1{,}2] \cup [2{,}6; +\infty[$




## 3. Utiliser, pour aller plus loin, parmorceaux(




La calculatrice offre une puissante commande de représentation conditionnelle des fonctions, dont on a déjà parlé dans la construction d'un tableau de signe.

Cette fonction peut être utilisée dans la résolution graphique des inéquations.

Placez-vous dans Y3. **`Parmorceaux(`** est accessible dans le menu **MATH** [math]

Choisir **`morceaux:1`** dans la fenêtre qui s'ouvre et recopier dans Y3 l'expression proposée ci-contre.

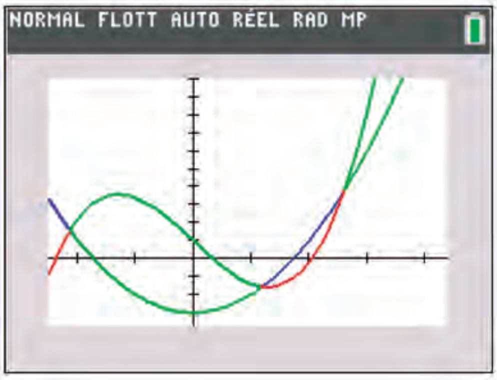
$\leq$ est accessible dans le menu **`tests`** de la calculatrice via [2nde] + [math]

Y1 et Y2 sont accessibles dans le menu **VAR Y** via [var]

N'oubliez pas de modifier, en vert, la couleur de la représentation.

Recommencez pour Y4.

Dans la fenêtre graphique, vous obtiendrez directement en vert, la zone répondant à notre inéquation.

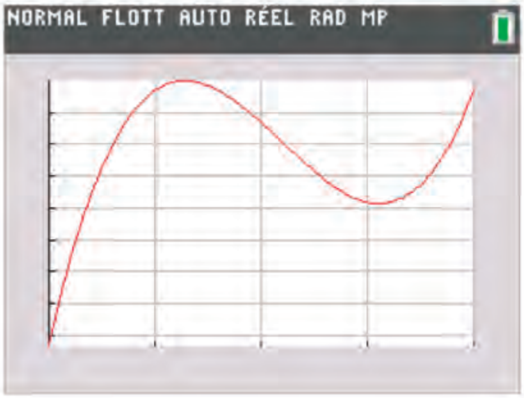


# Fonctions et applications

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## Enoncé




Une entreprise fabrique $x$ tonnes d'engrais vert avec $x \in [0; 20]$.

Le coût total de production en milliers d'euros est :

$C(x) = x^3 - 33x^2 + 300x + 69.$

1. Représenter graphiquement la fonction $C$ à l'aide de votre calculatrice. On affichera la grille en prenant $Xgrad = 5$ et $Ygrad = 100$.

2. Déterminer par une lecture graphique le coût en euros de la fabrication de 10 tonnes d'engrais.

3. On admet que le bénéfice $B(x)$, exprimé en milliers d'euros pour $x$ tonnes d'engrais fabriquées et vendues est :

$B(x) = -x^3 + 33x^2 - 216x - 69$ avec $x \in [0; 20]$.

Déterminer le maximum de $B$ sur l'intervalle $[0; 20]$.

4. Déterminer graphiquement à partir de quelle valeur de $x$ le bénéfice sera supérieur à 500.000 €.

## 1. Représentation graphique du coût

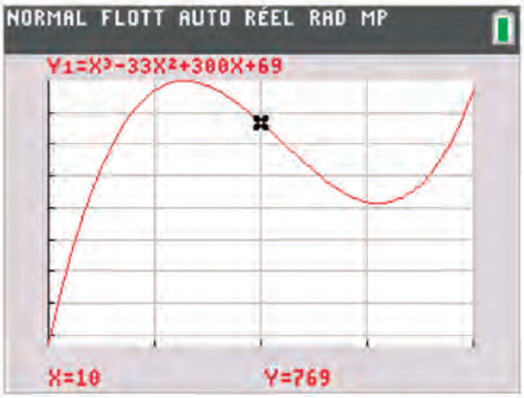

On entre l'expression de la fonction $C$ en appuyant sur f(x) puis on paramètre la fenêtre (touche fenêtre) en entrant l'ensemble de définition de la fonction, ici $x \in [0; 20]$ (il faut modifier les valeurs de **Xmin** et **Xmax**, voir écran ci-contre).

On ajuste la fenêtre automatiquement en appuyant sur zoom **0:AjustZoom**.

Pour afficher la grille on appuie sur 2nde zoom (format f3) et on choisit **LigneAff**

On trouve la représentation graphique située en haut de la page.

## 2. Lecture graphique du coût




Le coût en euros de la fabrication de 10 tonnes d'engrais est $C(10)$, on lit graphiquement cette valeur en utilisant 2nde trace (calculs f4) puis image et entrer 10.

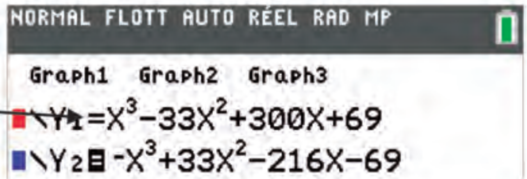
On trouve un coût de 769 000 €.

On peut aussi calculer directement l'image de 10 en écrivant **Y1(10)**. On rappelle que **Y1** est accessible dans var onglet **VAR Y** puis **Fonction…**

On trouve le même résultat : 769 000€.

## 3. Recherche du maximum

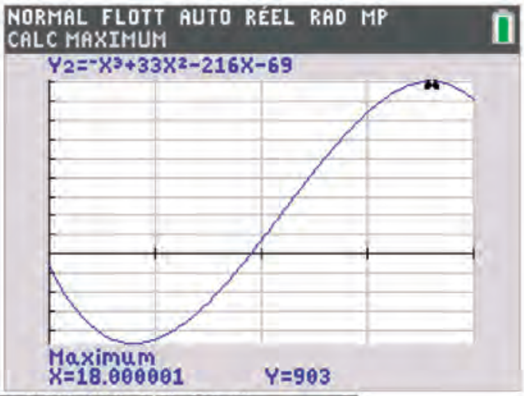
Commençons par entrer l'expression de la fonction $B$ dans Y2 en appuyant sur f(x). On désactive l'affichage de la fonction $C$ (qui correspond à Y1) en désélectionnant le fond noir du symbole =



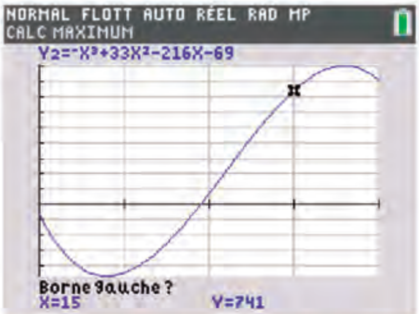

On affiche notre fonction dans une fenêtre calculée automatiquement en appuyant sur zoom `0:AjustZoom`.

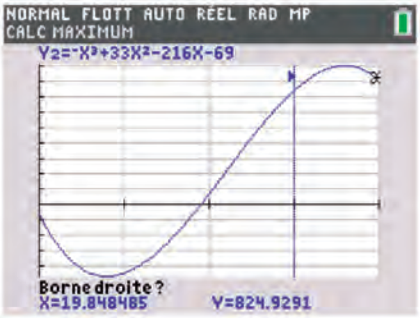
On détermine graphiquement le maximum de la fonction $B$ sur $[0;20]$ en appuyant sur 2nde trace (calculs f4).

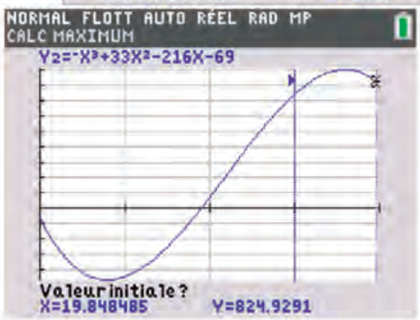
- On place son curseur un peu avant le maximum et on valide en appuyant sur entrer pour définir la borne inférieure.
- Puis on place son curseur un peu après le maximum (pour la borne supérieure) et on valide.
- On valide à nouveau pour la valeur initiale.




On trouve un bénéfice maximum d'environ 903 000€ pour la fabrication et la vente de 18 tonnes d'engrais.



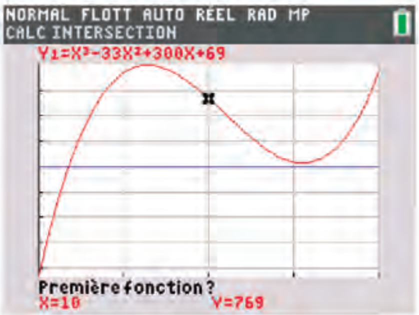



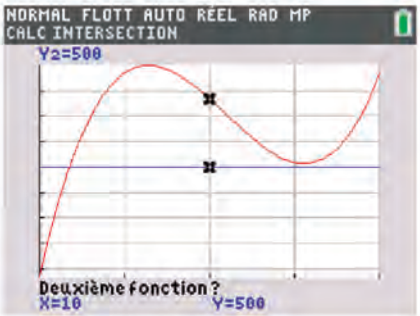



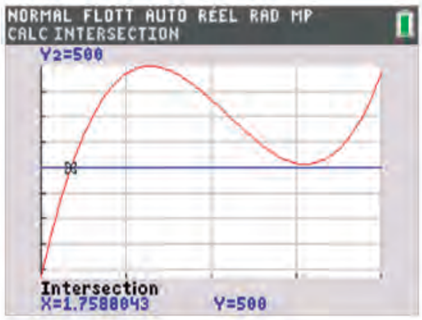

## 4. Recherche de profit

Pour déterminer graphiquement à partir de quelle valeur de $x$ le bénéfice sera supérieur à 500.000 €, on commence par tracer la droite d'équation $y = 500$ en écrivant dans f(x) **Y2=500**.

Puis on appuie sur graphe pour visualiser le graphique. Dans 2nde trace (calculs f4) on sélectionne `Intersection`.




Après avoir sélectionné les deux courbes des fonctions dont on cherche une intersection on trouve qu'à partir d'environ 1,759 tonne l'entreprise fait plus de 500 000€ de bénéfice.

Thème: Les applications de la calculatrice

# Résolution d'équation et simulations

TI-82 Advanced Edition Python

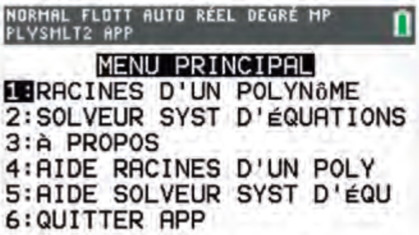
TI-83 Premium CE Edition Python

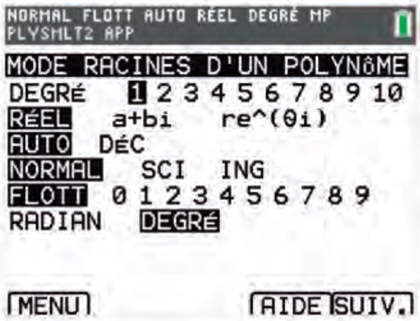
## Equation de degré 1

On souhaite résoudre l'équation $(E)\colon \frac{3}{4}x - \frac{5}{7} = 0$

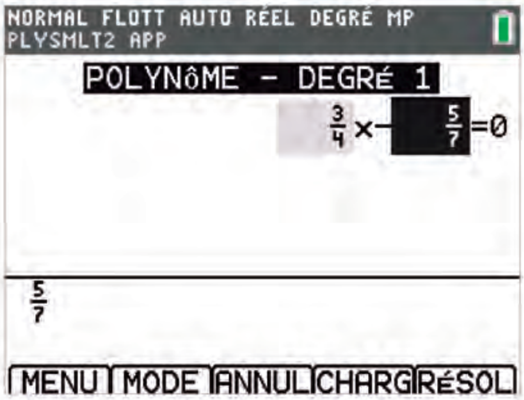
On peut utiliser une des applications présentes dans la calculatrice pour résoudre cette équation.

Appuyer sur [résol] puis sélectionner **PlySmlt2** et choisir **RACINE D'UN POLYNÔME** et sélectionner **DEGRÉ 1**.

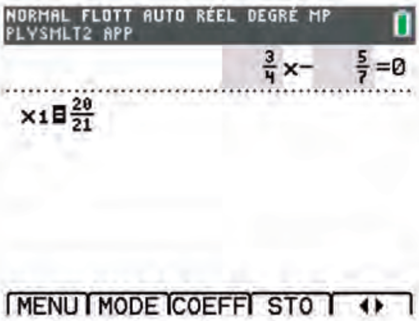



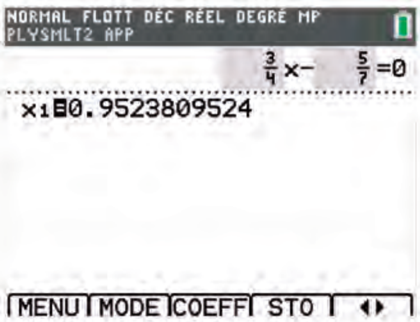




Il faut maintenant entrer les coefficients (voir ci-contre) puis, pour obtenir les solutions, on appuie sur **RÉSOL** (touche [graphe]).




Si on souhaite avoir une valeur approchée de la solution, on appuie sur [◀▶] (touche [graphe]).




**Conclusion** : Soit $S$ l'ensemble des solutions de l'équation $(E)$ alors on a

$$S = \left\{\frac{20}{21}\right\}$$

Pour quitter cette application, appuyer sur [2nde] [mode] (quitter) puis choisir **6:QUITTER APP**.

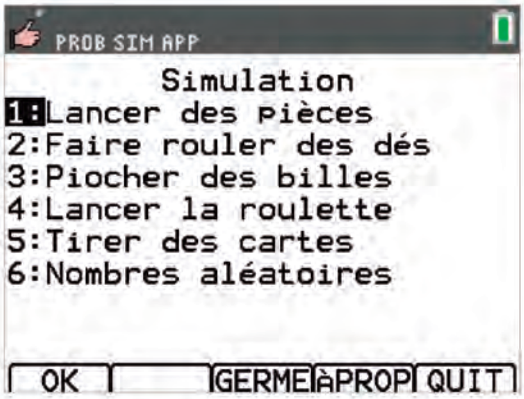
# Résolution d'équation et simulations

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Simulation aléatoire

Pour accéder à l'application de simulation, appuyer sur 2nde résol puis sélectionner tout en bas `0:Prob Sim`.

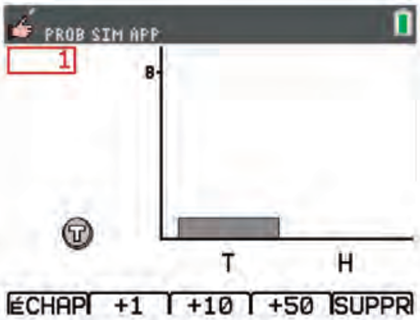


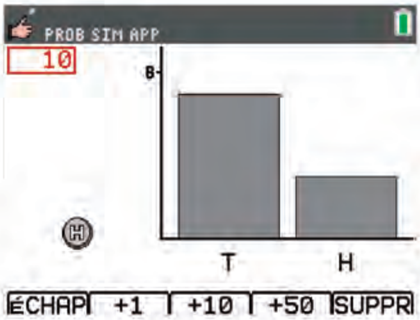

## Lancer des pièces

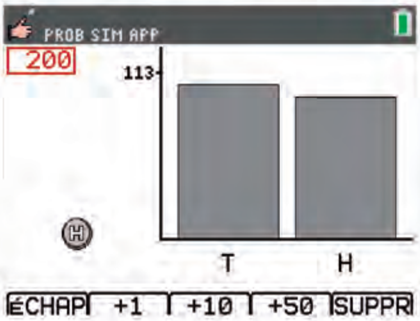
Choisir `1:Lancer` des pièces et appuyer sur **OK** (touche f(x)).

Pour lancer la pièce, appuyer sur **LANCE**. On peut recommencer le lancer en appuyant sur **+1** ou **+10** pour lancer la pièce 10 fois ou encore **+50** pour la lancer 50 fois.

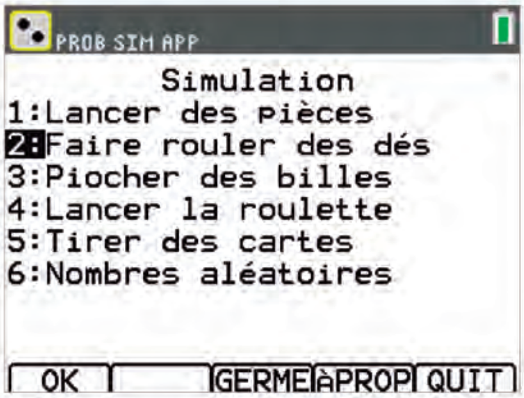
T est la première lettre du mot anglais « tail » qui signifie face en français et H est la première lettre de « head » pour pile.



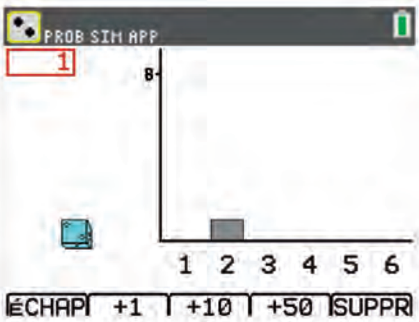

Pour connaitre l'effectif du nombre de piles et faces il suffit de naviguer avec les flèches de direction. Pour sortir de la simulation appuyer sur **ÉCHAP**.

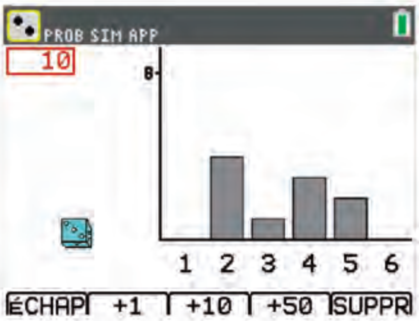


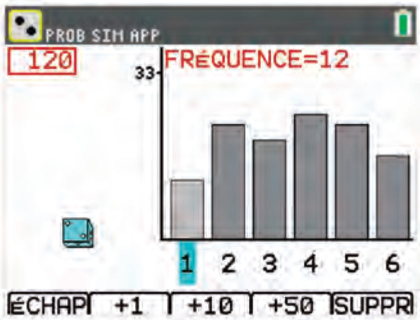

## Faire rouler des dés

Pour la simulation de lancers de dés, choisir `2:Faire rouler des dés` et appuyer sur **OK** (touche f(x)).

Les commandes sont les mêmes que pour la simulation précédente.



# S'approprier l'éditeur de listes

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## Contexte

L'éditeur de listes est un outil puissant de la calculatrice pour nous aider à traiter des données, à les représenter et à calculer des indicateurs statistiques.

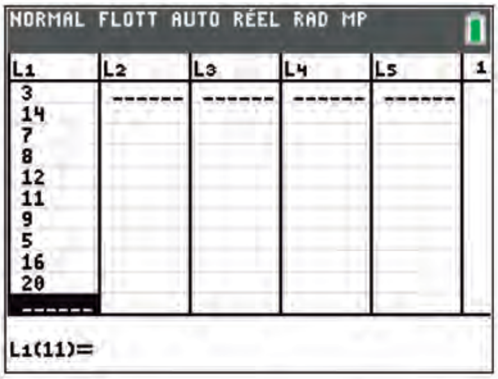
Etudions, pour commencer, la série de notes suivantes : 3,14,7,8,12,11,9,5,16,20

## 1. Saisir une série statistique


On accède à l'éditeur de listes à l'aide de la touche stats

On sélectionne l'item **1:Modifier...** ce qui va nous permettre d'accéder à l'éditeur de données.

Dans ce premier exemple, il s'agit de saisir chacune des notes.




On positionne, éventuellement à l'aide des flèches de la calculatrice, le curseur sur la première ligne de $L_1$.

En cas d'erreur de saisie, il est possible, toujours avec les flèches de la calculatrice, de revenir sur la ligne pour la corriger.

On valide chaque saisie à l'aide de la touche entrer

## 2. Trier dans l'ordre croissant


Une fois la saisie réalisée, il est possible de demander à l'éditeur de trier les données dans l'ordre croissant pour en faciliter ultérieurement l'étude.

Nous utilisons la fonction **TriA(** dans le menu accessible via la touche stats

La calculatrice nous renvoie vers l'écran de calcul.

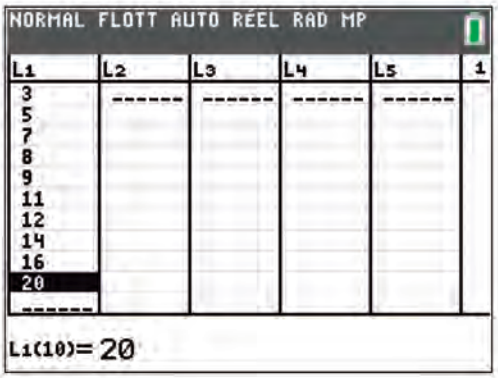
On complète la saisie avec le nom de la liste que l'on souhaite ordonner (ici $L_1$) à l'aide de la combinaison de touches 2nde + 1

$L_1$ Obtenu via le bon appel de touche. Le 1 est en indice

L1 Saisie erronée - ne marchera pas

Il est important de bien comprendre que l'appel des listes ne se fait surtout pas en tapant **L** puis **1**. Il s'agit d'un nom de variable spécial qu'il faut sélectionner directement. Sur la calculatrice, les noms de listes sont situés en bleu sur les touches 1,2,3,4,5 et 6 pour $L_1$, $L_2$, $L_3$, $L_4$, $L_5$ et $L_6$

On valide l'instruction avec entrer



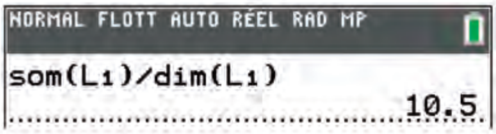

Un retour dans l'éditeur de listes nous montre $L_1$ désormais classée dans l'ordre croissant.

# S’approprier l’éditeur de listes



## 3. Calculer individuellement des indicateurs

Par exemple, la moyenne de notre série de notes s’obtient en divisant la somme des valeurs par l’effectif total de la série.

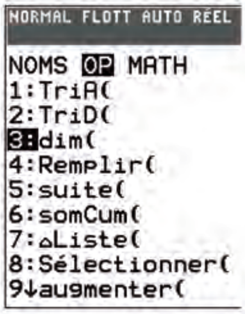


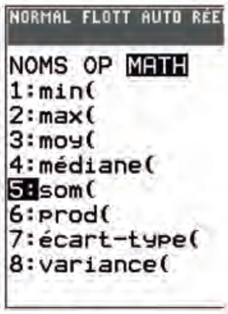

Pour cela, on peut opérer à partir de l’écran de calculs de différentes façons.

On accède à des fonctions mathématiques dédiées, dans le menu **listes**, à l’aide de la combinaison de touches `2nde` + `stats`

Ainsi **dim(** permet d’obtenir, dans le menu **OP**, le nombre d’éléments contenus dans la liste désignée en paramètre.

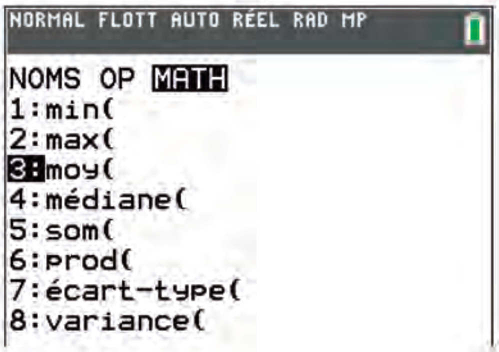
**som(** permet d’obtenir, dans le menu **MATH**, la somme des éléments contenus dans la liste désignée en paramètre.

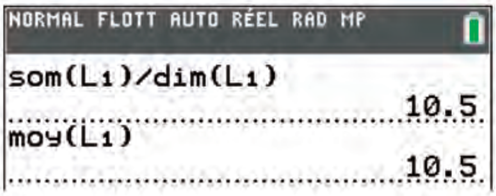



On peut également obtenir directement un indicateur.

Par exemple, pour la moyenne, on utilise fonction **moy(**



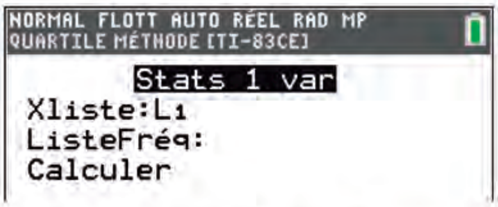




## 4. Calculer plusieurs indicateurs

La calculatrice permet d’obtenir un résumé de l’ensemble des indicateurs d’une série statistique.


Même si on va y revenir en détails par la suite, testons cela déjà une première fois avec notre série, contenue dans **L1**

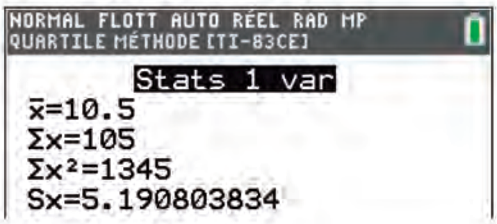
Pour cela, on utilise la touche `stats` et on sélectionne dans le menu **CALC**, la fonction **Stats 1 Var**.




On configure de la manière ci-contre. **ListeFréq** est laissé vide.

On obtient notre résumé, une fois validé en se déplaçant sur **Calculer** et en appuyant sur la touche `entrer`

On comprend que $\bar{x}$ représente la moyenne mais pour en apprendre davantage, poursuivons notre lecture.



# S’approprier l’éditeur de listes

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

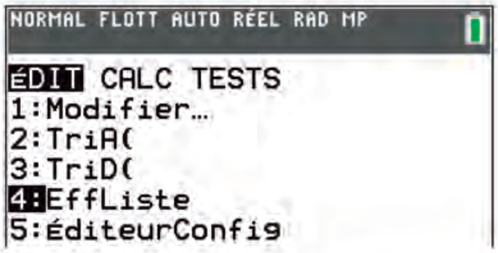
## Contexte

Etudions une nouvelle série de notes dans laquelle deux élèves ont eu 8, un élève a eu 9, 3 élèves ont eu 10, deux élèves ont eu 12 et enfin deux autres élèves ont eu 13.

## 1. Saisir une série « valeurs - effectifs »

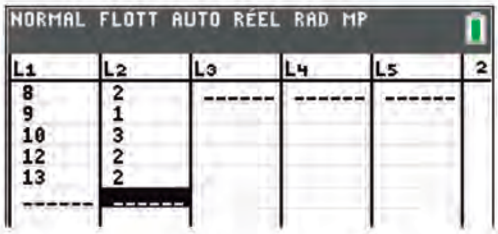
On accède à l’éditeur de listes à l’aide de la touche stats

On commence par effacer le contenu de la liste L1 précédente à l’aide de l’item **4:EffListe** et on complète l’instruction dans l’écran avec le nom de la liste.




Attention à bien utiliser la combinaison de touche 2nde + 1 pour saisir le nom de la liste L1

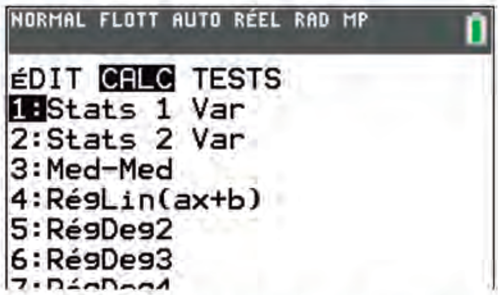

On reviendra sur les différentes façons de gérer l’effacement du contenu de nos listes dans la fiche suivante « 13 - Générer des données ».




Dans L1, on saisit les différentes valeurs distinctes, dans l’ordre croissant, de notre nouvelle série.

Dans L2, on saisit les effectifs correspondants.

## 2. Calculer automatiquement des indicateurs




Calculons simultanément les indicateurs statistiques pour notre série.

Pour cela, on utilise la touche stats et on sélectionne dans le menu **CALC**, la fonction **Stats 1 Var**.

Dans la fenêtre de configuration qui s’ouvre, on précise alors que les valeurs sont contenues dans L1 à l’aide des touches 2nde + 1


Les effectifs sont contenus dans L2 ( 2nde + 2 ).

Le terme **ListeFréq** peut induire en erreur avec la notion de fréquence mais il s’agit bien de désigner la liste des effectifs.

Enfin, on valide en se déplaçant avec les flèches sur **Calculer** puis entrer

# S’approprier l’éditeur de listes



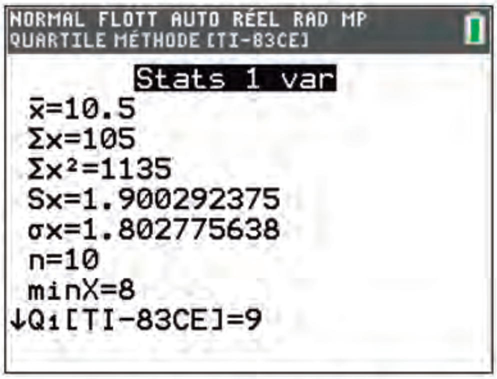
La calculatrice nous propose alors un résumé de l’ensemble des indicateurs statistiques usuels.

$\bar{x}$ représente la moyenne de notre série statistique.

$\sum x$ représente la somme des valeurs, en tenant compte des effectifs renseignés dans `L2`

$n$ est l’effectif total

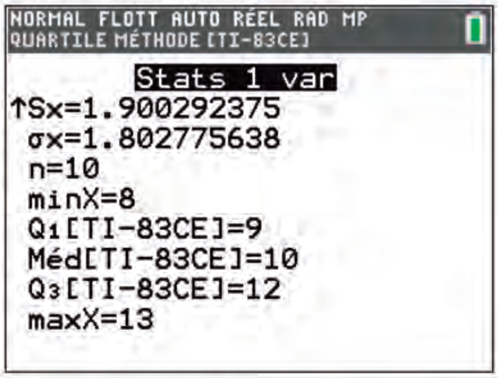
$\sigma x$ représente l’écart-type (à ne pas confondre avec l’indicateur $Sx$ qui est une estimation de l’écart-type de la population dont serait extraite la série étudiée).




En utilisant les flèches de la calculatrice (⌃ ou ⌄), on atteint d’autres indicateurs comme la valeur minimale $minX$ ou maximale $maxX$.

Enfin on trouve également les quartile $Q_1$ et $Q_3$ ainsi que la médiane $Méd$.

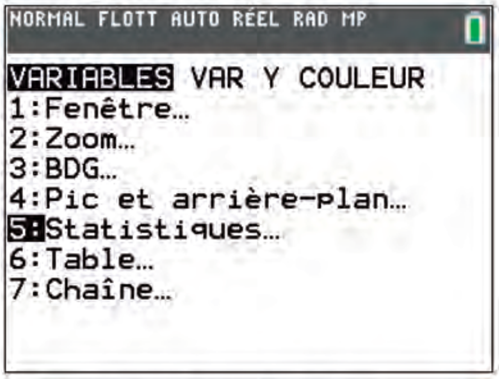
On quitte le résumé en appuyant sur annul ou à l’aide de la combinaison de touches 2nde + mode



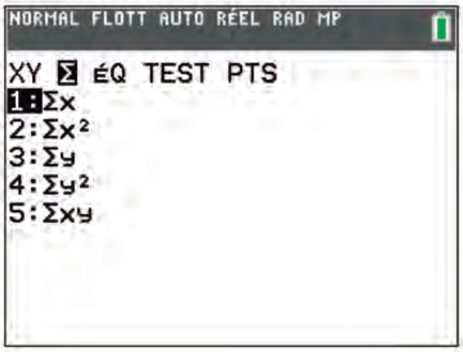

Les indicateurs sont automatiquement sauvegardés et réutilisables dans l’écran de calculs.

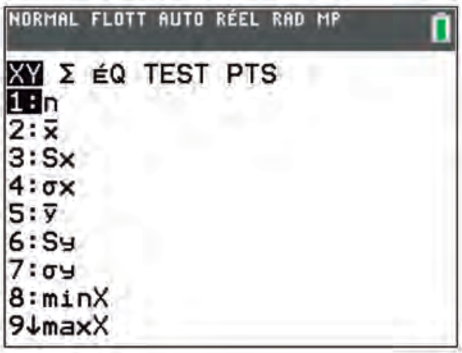
Pour cela, on appuie sur la touche var puis on choisit l’item **Statistiques…**

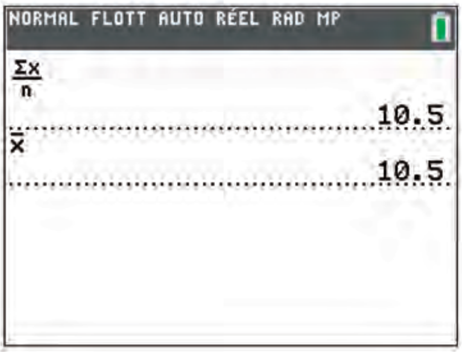
On sélectionne ensuite l’indicateur désiré, éventuellement en naviguant si besoin dans les différents onglets.




On peut, par exemple, retrouver la moyenne de la série en effectuant le quotient de la somme des valeurs par l’effectif total.



# Générer des données

TI-82 Advanced
Edition Python

TI-83 Premium CE

Edition Python

## Contexte

On poursuit la découverte de l'éditeur de listes. Dans cette partie, on va générer des données (aléatoires ou pas) pour pouvoir ensuite tester différentes représentations graphiques.

## 1. Nettoyer nos données

On commence par effacer l'ensemble des données contenues dans notre éditeur de listes. On peut procéder de plusieurs façons.

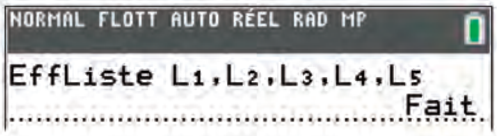

Si on souhaite effacer toutes les listes d'un seul coup, on peut se rendre dans le menu **MéMOIRE** à l'aide de la combinaison de touches 2nde +


On sélectionne l'item **4:EffTtesListes**. Une fois l'opération validée dans l'écran de calculs, les listes sont nettoyées de leurs contenus.

On peut également cibler les listes dont on souhaite effacer le contenu.

Dans ce cas, on se rend dans le menu **stats** stats

On utilise la commande **EffListe** complétée du nom des listes à effacer (dans l'exemple L1 à L5) séparée par des virgules ,



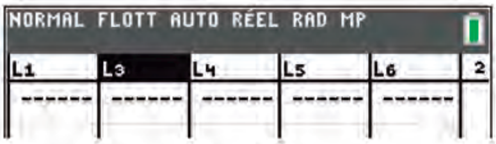

On peut également procéder directement dans l'éditeur de listes.

Il suffit pour cela de se positionner sur l'entête de la liste que l'on souhaite effacer. Puis on appuie sur la touche annul et enfin on valide avec la touche entrer

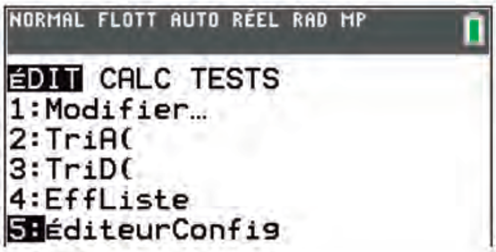


Il arrive fréquemment de confondre la touche annul et la touche suppr lors de cette manipulation. Cela a pour effet de faire disparaitre la colonne contenant la liste sur laquelle on s'était positionné.

Dans la capture ci-contre, L2 a disparu de l'éditeur de listes.

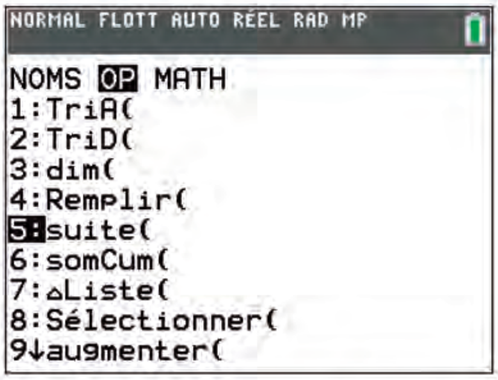



On peut la faire réapparaitre à l'aide de la commande **EditeurConfig** dans le menu **Stats**. Une fois l'instruction validée, L2 réapparaitra dans l'éditeur de listes.

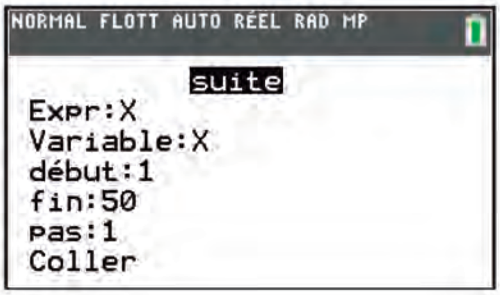



## 2. Générer une liste de nombres entiers

Lorsqu'on manipule des données, il n'est pas rare d'avoir besoin de les associer à un nombre entier, qui peut représenter un numéro de tirage, en probabilité, par exemple. Bien sûr, on pourrait saisir à la main, chacun des numéros mais cela peut s'avérer rapidement fastidieux si ce nombre est important.



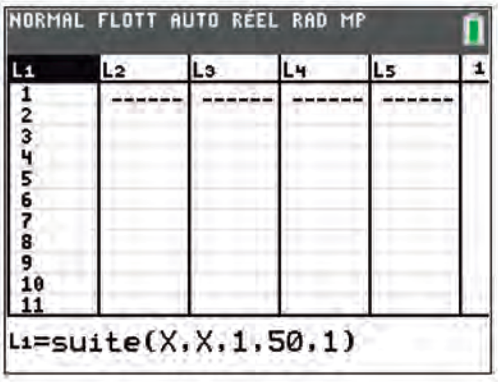

Plaçons le curseur dans l'entête de colonne **L1** et, à l'aide de la combinaison de touches [2nde] + [stats], on se rend dans l'onglet **OP** puis on sélectionne la commande **suite(** .




Ceci a pour effet de nous renvoyer vers une fenêtre de saisie, que l'on remplit de la manière ci-contre.

Dans l'expression algébrique $x$, on demande à la calculatrice d'incrémenter la valeur de $x$ par pas de 1, en partant de 1 et pour aller jusqu'à 50.

On déplace le curseur sur **Coller** et on valide [entrer]




Ceci a pour effet de générer les nombres entiers de 1 à 50.

Pour générer une liste d'entiers jusqu'à 100, il suffit de remplacer la valeur de fin par 100.

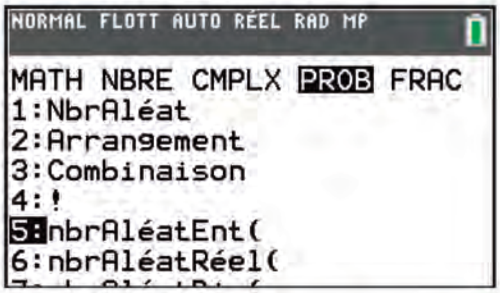
On pourrait générer les carrés des nombres entiers en remplaçant l'expression X par X² dans la fenêtre de saisie.

Pour générer une liste de nombres impairs, il suffit de remplacer la valeur du pas par 2.

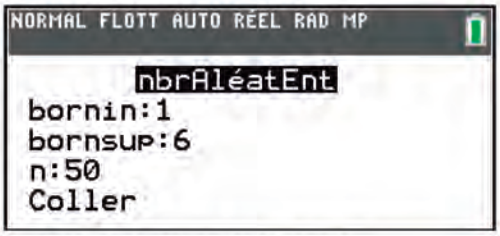
# Générer des données



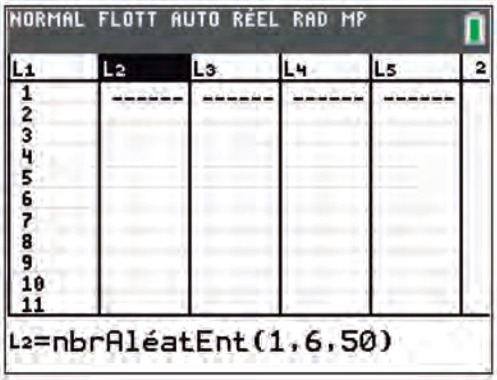
## 3. Générer une liste de nombres aléatoires




Plaçons le curseur dans l'entête de colonne **L2** et, à l'aide de la touche [math], on se rend dans l'onglet **PROB** puis on sélectionne la commande **nbrAléatEnt(** .

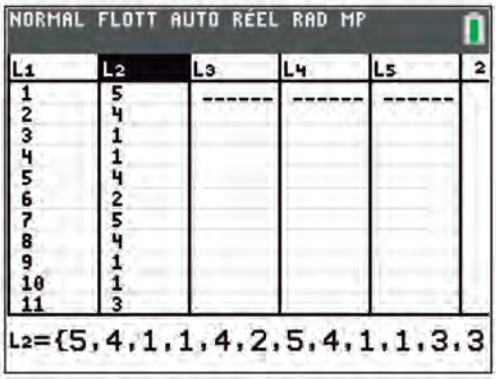


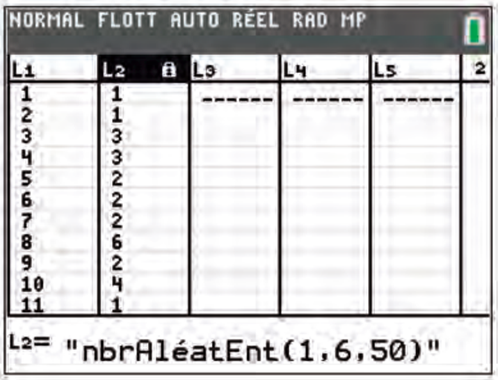

Ceci a pour effet de nous renvoyer vers une fenêtre de saisie, que l'on remplit de la manière ci-contre pour obtenir une liste de 50 nombres entiers aléatoires compris entre 1 et 6 (ce qui pourrait correspondre à 50 tirages d'un dé équilibré à 6 faces)




Une fois **Coller** puis la saisie d'en l'entête de colonne **L2** validée, on obtient notre liste de nombres aléatoires.

Mais la manipulation n'est valable qu'une seule fois et si l'on souhaite renouveler l'expérience, il faut recommencer la saisie dans l'entête de **L2** depuis l'onglet **PROB**




Il existe un moyen d'éviter cela lors de la première saisie pour demander à la calculatrice de mémoriser la ligne de commande. Il faut saisir l'expression entre guillemets à l'aide de la combinaison de touches [alpha] [+]

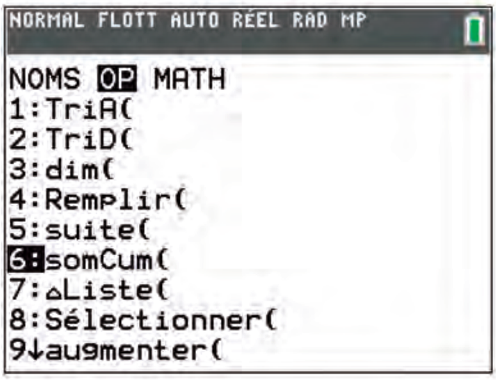
Ainsi, il suffira de retourner dans l'entête de colonne et de modifier la ligne mémorisée (par exemple de ressaisir 1). Ceci aura pour effet de faire comprendre à la machine que l'on souhaite réaliser un nouveau tirage de 50 nombres entiers aléatoires sans avoir à recommencer toute l'opération de saisie depuis la fenêtre de paramétrage de la fonction **nbrAléaEnt(**

# Générer des données

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## 4. Générer une liste à partir d'une liste définie

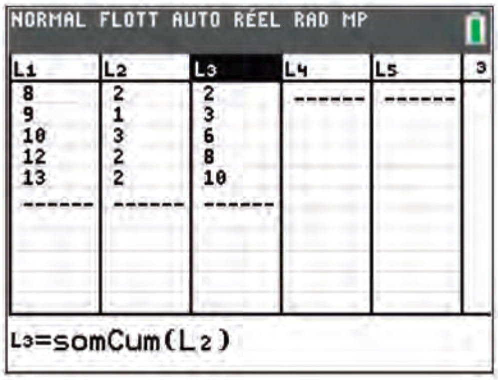



La calculatrice offre la possibilité de traiter les données existantes pour générer de nouvelles listes.

Par exemple, pour retrouver la valeur de la médiane, on souhaite dresser la liste des effectifs cumulés croissants, dans `L3`, à partir de la liste des effectifs contenus dans `L2`.

Pour cela, dans l'éditeur de listes, on place le curseur dans l'entête de colonne `L3`.

On sélectionne la fonction `somCum(` dans l'onglet `OP` du menu `Listes` [2nde] [stats] et on complète avec la liste `L2` passée en paramètre. Une fois la commande validée, la calculatrice complète la liste `L3` avec les effectifs cumulés attendus.




On souhaite augmenter toutes les notes de 50%, il faut donc multiplier chaque valeur par $CM = 1 + \frac{50}{100} = 1{,}5$

On peut saisir dans l'entête de colonne de `L4`, la formule suivante :

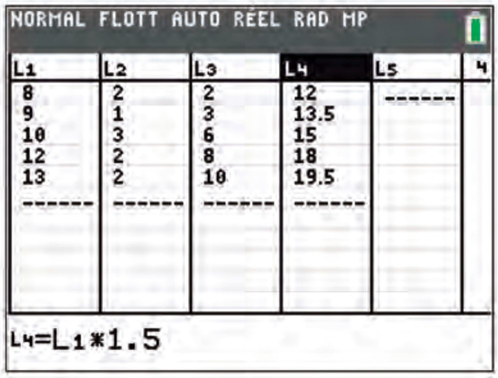
`L4 = L1*1.5`

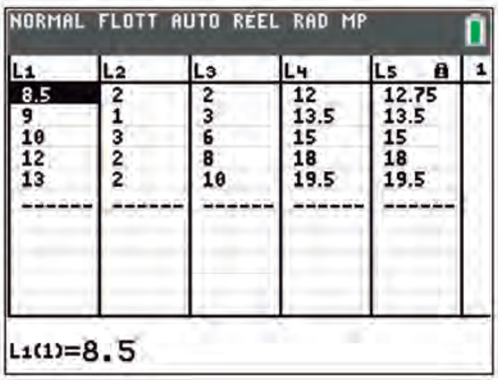
Les valeurs sont automatiquement calculées.

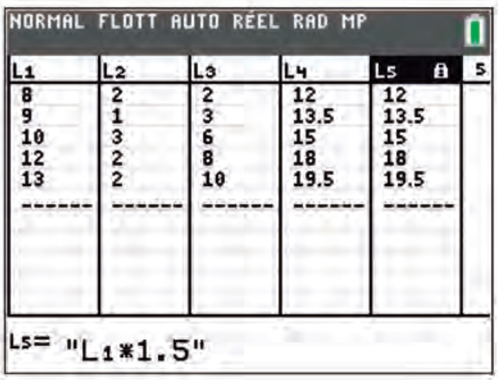
Enfin il est possible de saisir la formule entre guillemets.

`L5 ="L1*1.5 "`

Cela a pour effet de lier dynamiquement les deux listes (affichage d'un cadenas) et toute modification dans la liste de départ est recalculée dans la liste d'arrivée. Comparez `L4` et `L5` dans le dernier exemple. Si l'on modifie la note de 8 par 8.5, on voit que la valeur dans `L5` est ajustée (12,75) tandis qu'elle est maintenue à 12 dans `L4`.



# Médiane, écart-type, quartiles



## Trouver des indicateurs statistiques

Madeleine fabrique des petits pains d'une masse d'environ 80g. Voici un relevé statistique de sa dernière production :

| Masse (en grammes) | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 86 | 87 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Effectifs | 4 | 11 | 17 | 45 | 68 | 51 | 39 | 120 | 114 | 68 | 41 | 31 | 21 | 6 |

1. Quel est le caractère étudié ?

2. Entrer ces valeurs dans la calculatrice dans la liste `L1` pour les masses et dans la liste `L2` pour les effectifs associés.

3. A l'aide de la calculatrice, déterminer l'effectif total, la moyenne, l'écart type, la médiane et les quartiles de cette série statistique.

4. A l'aide des résultats de la question 3. Compléter les phrases suivantes :

- « Madeleine a fabriqué un total de ……. petits pains. »
- « Environ 50% des petits pains ont une masse inférieure à ……. »
- « Environ 25% des petits pains ont une masse supérieure à ……. »

5. Pour décorer les petits pains de Madeleine, Anne a ajouté une pépite de chocolat de 5 grammes sur chacun d'eux.

   a) Quelle liste faut-il modifier pour tenir compte de ce changement de masse ? Effectuer ce changement.

   b) A l'aide de la calculatrice, déterminer la nouvelle masse moyenne des petits pains ainsi que l'écart type ? Pouvions-nous prévoir ces résultats sans utiliser la calculatrice ?

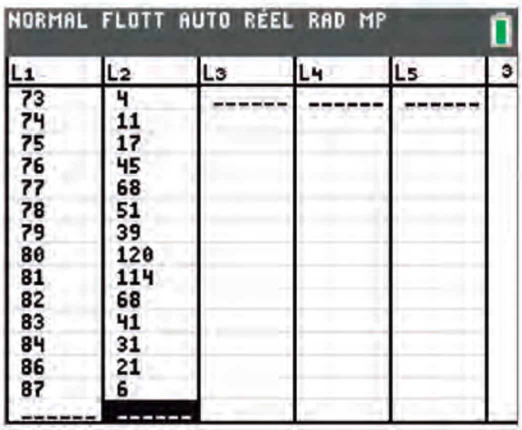
## 1. Caractère étudié

Le caractère étudié est la masse des petits pains (exprimée en grammes).

## 2. Entrer les valeurs dans sa calculatrice

On appuie sur [stats] **`Calc`** pour entrer les valeurs du caractère (dans `L1`) et les effectifs associés (dans `L2`).

On effacera éventuellement le contenu d'anciennes listes présentes. On pourra pour cela se référer aux fiches précédentes.



# Médiane, écart-type, quartiles

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## 3. Indicateurs statistiques

Pour obtenir les différents indicateurs statistiques, on appuie sur [stats] onglet **CALC** puis sélectionner **Stats 1 Var**.

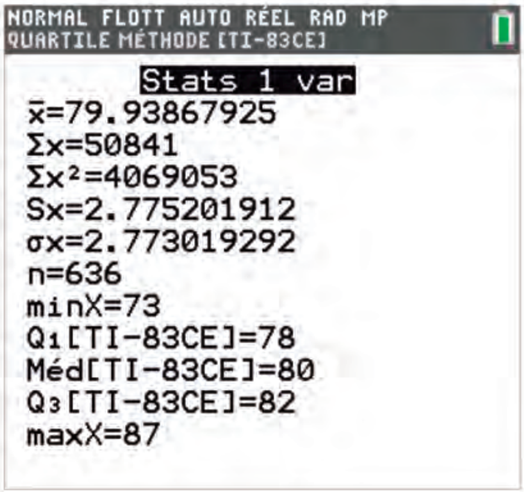
Dans **Xliste** on entre **L1** (touche [2nde] [1]), la liste contenant les valeurs du caractère et dans **ListeFréq** on entre **L2**, la liste des effectifs.

Puis on sélectionne **Calculer** et on valide.

La moyenne est donc $\bar{x} \approx 79{,}939$ grammes.

L'écart type est $\sigma \approx 2{,}773$.

La médiane est $Me = 80$, le premier quartile $Q_1 = 78$ et le troisième quartile est $Q_3 = 82$.


## 4. Compléter les phrases

Pour compléter la première phrase, il faut utiliser la médiane $Me = 80$.

Ainsi « Environ 50% des petits pains ont une masse inférieure à 80 ».

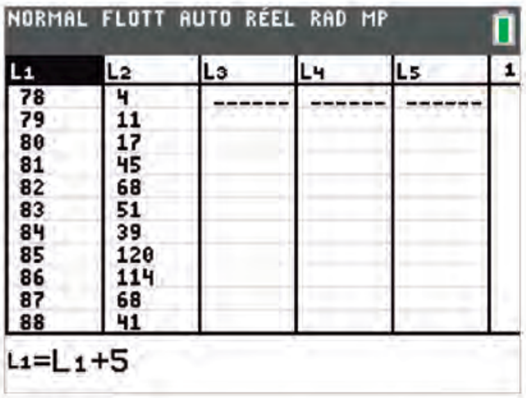
La seconde phrase va utiliser le troisième quartile $Q_3 = 82$ :

« Environ 25% des petits pains ont une masse supérieure à 82 ».

Et enfin l'effectif total est $n = 636$ donc « Madeleine a fabriqué un total de 636 petits pains. »

## 5. a. La pépite en chocolat

Il faut modifier la liste **L2**. Pour ajouter rapidement 5 à toutes les valeurs de la liste on se place tout en haut de **L2** et on écrit **L2+5** puis on valide.




## 5. b. Nouvelles valeurs des indicateurs

Pour refaire les calculs, on appuie sur [stats] onglet **CALC** puis sélectionner **Stats 1 Var** et valider plusieurs fois. On obtient les résultats ci-contre :

La nouvelle moyenne est de 84,939 grammes et l'écart type est de 2,773.

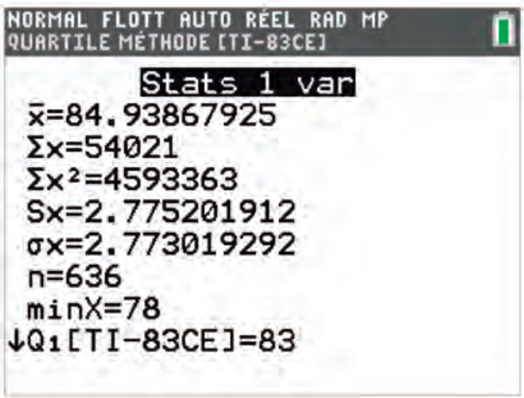
On pouvait trouver ces valeurs à l'aide du cours :

**Propriété** : Soit $m$ la moyenne d'une série statistique. Si toutes les valeurs de la série augmente de $a$ (avec $a \in \mathbb{R}$) alors la nouvelle moyenne est $m + a$.

L'ancienne moyenne était de $\bar{x} \approx 79{,}939$ donc après avoir augmenté de 5 grammes chaque petits pains, la nouvelle est de $79{,}939 + 5 = 84{,}939$.

**Propriété** : Soit $\sigma$ l'écart type d'une série statistique. Si toutes les valeurs de la série augmente de $a$ (avec $a$ un réel) alors le nouvel écart type n'est pas modifié, il vaut toujours $\sigma$.

D'après le cours l'écart type n'est pas modifié.

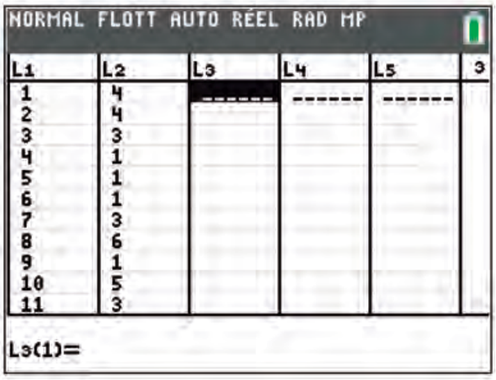


# Représentation graphique de données

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

## Contexte

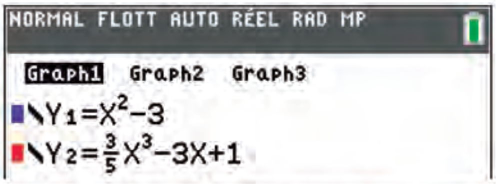



On termine notre prise en main de l'éditeur de listes pour tester différentes représentations graphiques à partir d'un jeu de données existant.

On dispose d'une liste de nombres entiers de 1 à 50 dans `L1` et de 50 nombres entiers aléatoires compris entre 1 et 6 dans `L2`.

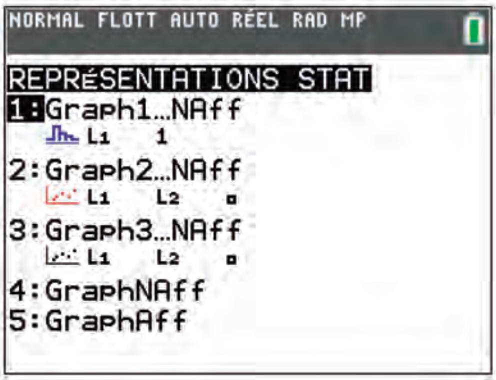
## 1. Préparer la représentation graphique

Pour utiliser la représentation graphique de données statistiques, il faut prendre l'habitude de commencer par se rendre dans l'éditeur de fonctions pour désactiver la représentation graphique des différentes courbes représentatives.




Pour cela, on appuie sur [f(x)] et on s'assure que tous les signes « = » sont sur fond blanc (voir la fiche « 4 - représentation graphique de fonctions »).

Cette démarche réalisée, on peut maintenant se rendre dans l'éditeur de représentation statistique à l'aide de la combinaison de touches [2nde] + [f(x)]

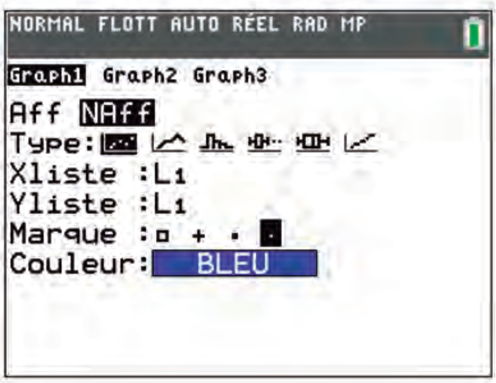



C'est dans cet éditeur que nous allons pouvoir configurer la représentation graphique souhaitée pour nos données.

## 2. Représenter un nuage de points

La représentation sous forme de nuage de points est souvent un point de départ lorsqu'on cherche une corrélation entre deux variables statistiques ou bien lorsqu'on souhaite évaluer visuellement la répartition d'une valeur dans une série statistique.

Nous pouvons configurer jusqu'à trois représentations statistiques simultanément.




Nous commençons par utiliser la première **`Graph1`** en positionnant le curseur dessus et en appuyant sur la touche [entrer]

Examinons les différentes options proposées par l'éditeur.

Il faut activer la représentation en sélectionnant l'option **`Aff`**. Elle apparait sur fond noir **`Aff`**

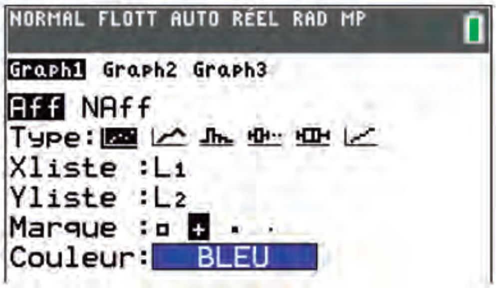
La représentation sous forme de nuage de points est la première icône

Il faut ensuite spécifier à la calculatrice le nom de la liste contenant les abscisses (**`Xliste`**) puis le nom de la liste contenant les ordonnées (**`Yliste`**).

On peut enfin changer la « forme » des points (**`Marque`**) et leur couleur de représentation.

# Représentation graphique de données

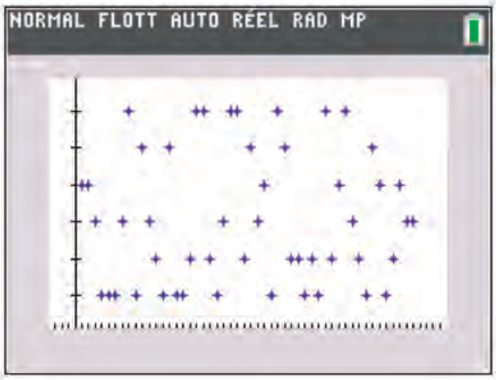





Ainsi, dans notre cas, nous souhaitons placer nos données aléatoires contenues dans **L2** en ordonnée. Le numéro de tirage contenu dans **L1** servira d'abscisse.

On appelle les listes à l'aide de la combinaison de touches 2nde 1 pour **L1** ou pour **L2** 2nde 2

On a donc configuré notre représentation graphique de la façon ci-contre.

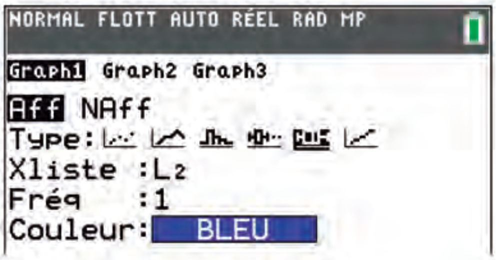



Pour obtenir notre représentation graphique, on commence par se rendre dans le menu **ZOOM** à l'aide de la touche zoom

En effet, il est fort probable que l'affichage soit configuré pour la représentation de fonctions. Il est donc utile de calibrer la fenêtre de représentation à l'aide de la fonction **ZoomStat**.

Une fois validé, la calculatrice nous bascule sur la représentation graphique voulue du nuage de points.

## 3. Représenter une boite à moustache




La représentation sous forme de boite à moustache est, dans sa réalisation, assez proche des manipulations précédentes.

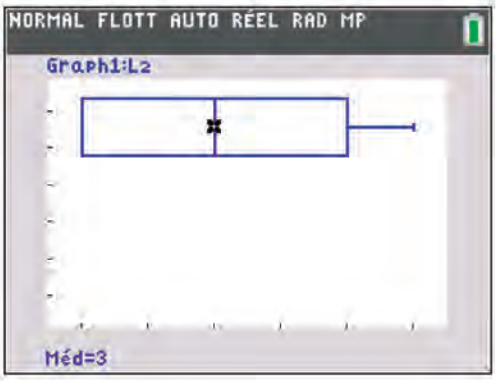
On se rend de nouveau dans l'éditeur de représentation statistique.

On souhaite de nouveau travailler avec **Graph1**.

Cette fois-ci nous sélectionnons le cinquième type de représentation

Nos données sont toujours contenues dans la liste **L2** et leur effectif respectif vaut 1.

Dans le cas d'une représentation type « valeur - effectif » où les effectifs seraient contenus dans **L3**, par exemple, on remplace alors 1 par **L3** (voir le point suivant « représenter un histogramme »).




Pour obtenir notre représentation graphique, on passe de nouveau par la fonction **ZoomStat** à l'aide de la touche zoom

L'appui sur la touche trace permet de lire sur le graphique les valeurs des indicateurs qui ont permis de construire la boite à moustache (Min, Q1, Médiane, Q3 et Max)

# Représentation graphique de données

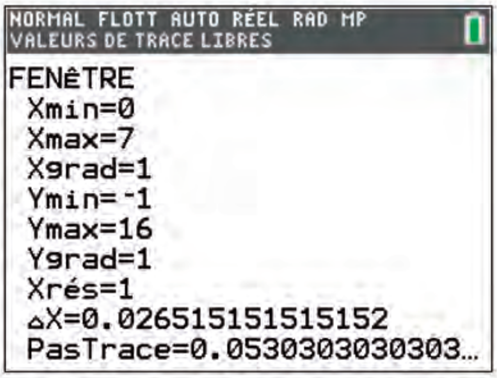


## 4. Représenter un diagramme en bâtons


Commençons par le cas où l'on ne dispose de données que dans une seule liste, ici `L2`.

Nous poursuivons notre travail avec **`Graph1`** et sélectionnons cette fois-ci la représentation sous forme « d'histogramme » (au sens graphique et non mathématique du terme).

Nous configurons notre représentation de la manière ci-contre.

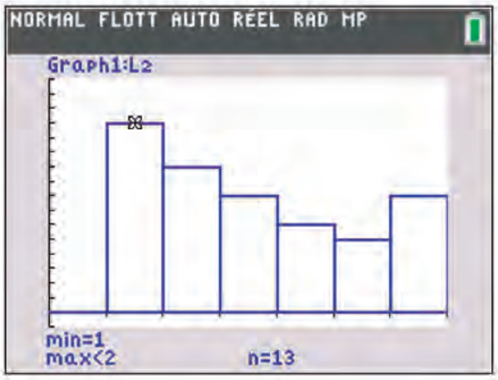



Cette fois-ci, pour obtenir notre représentation graphique, nous allons utiliser le menu **`fenêtre`** à l'aide de la touche [fenêtre]

Il faut alors veiller à configurer manuellement différents paramètres en fonction des données que l'on souhaite représenter.

Ici, nos données vont de 1 à 6 par pas de 1. On va donc configurer les paramètres de la manière ci-contre.

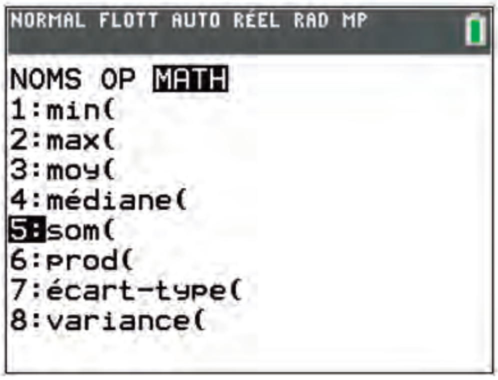
En particulier, on se montrera vigilant au paramètre **`XGrad`** afin qu'il respecte le pas entre nos valeurs (ici 1).




On accède alors à la représentation graphique à l'aide de la touche [graphe]

La touche [trace] nous permet de relever les effectifs établis par la calculatrice.

Dans l'exemple ci-contre, la calculatrice a détecté 13 fois le chiffre 1 dans la série contenue dans `L2`.



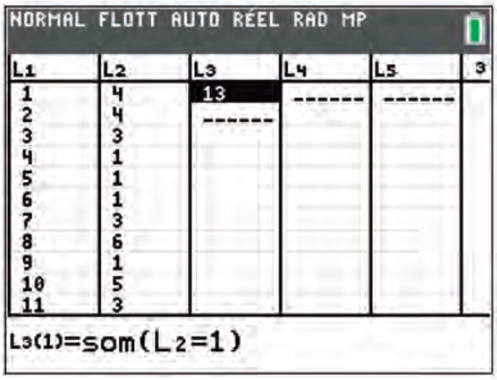

On peut alors s'interroger sur la possibilité de retrouver ce résultat par le calcul. C'est tout à fait possible.

Pour cela, on retourne maintenant dans l'éditeur de liste et on se positionne, pour commencer, dans la première cellule de L3 où l'on saisit la formule **`som(L2=1)`** où **`som(`** est dans l'ongle **`MATH`** du menu **`listes`** [2nde] [stats] et le signe = disponible dans le menu **`tests`** [2nde] [math]

# Représentation graphique de données



On retrouve « 13 » le résultat obtenu avec la fonction **trace** dans la représentation graphique. L'instruction permet de comptabiliser le nombre de « 1 » dans la liste **L2**.

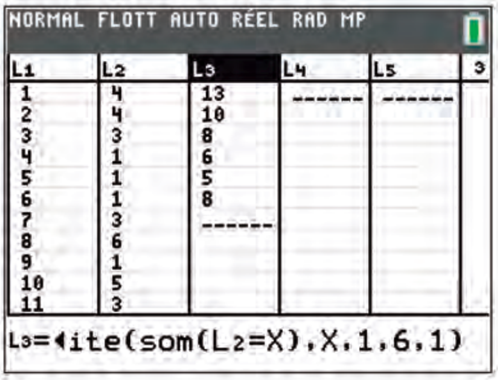


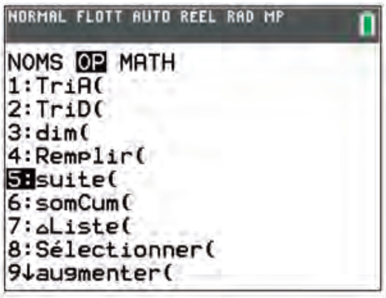

Si l'on combine tout ce que nous avons vu ensemble sur le thème des statistiques, on peut alors dresser le tableau complet des effectifs de la série étudiée, pour la représenter par la suite.

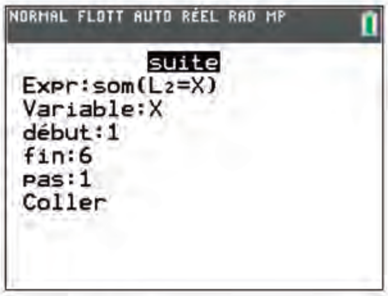
Pour cela, nous allons nous placer dans l'entête de **L3** et utiliser la fonction **suite** disponible dans l'onglet **OP** du menu **listes**.

Nous nous servons de la fenêtre de saisie qui s'ouvre pour obtenir l'instruction suivante : **suite(som(L2=X),X,1,6,1).**

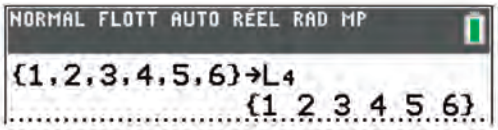
Une fois tout validé, nous obtenons la liste des effectifs des valeurs 1 à 6 contenues dans la liste **L2**.



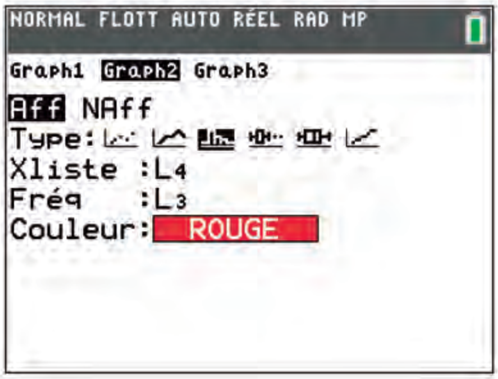



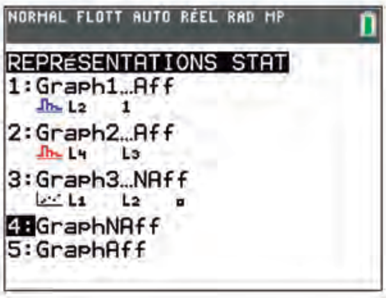

Nous créons une liste **L4** contenant les valeurs de 1 à 6

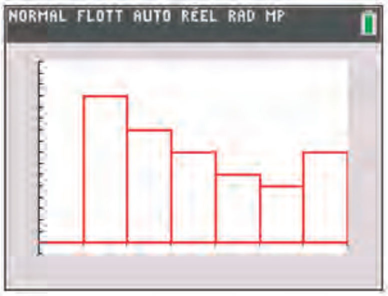



Nous configurons alors **Graph2** une nouvelle représentation graphique de données statistiques de la manière ci-contre et retournons dans la fenêtre graphique pour constater que les deux représentations se superpose parfaitement. On a bien retrouvé les effectifs recherchés.




On n'oublie pas pour finir de désactiver les représentations graphiques de l'éditeur statistique à l'aide de la fonction **GraphNAff**



# Diagramme en boîte



## Comparaison de deux séries

Deux clubs d'échec s'affrontent. On étudie l'âge des participants.

Pour le club A les âges des membres sont : 45 ; 51 ; 66 ; 19 ; 27 ; 23 ; 40 ;52 ;22 ; 26 ; 61 ; 62 ; 44 ;45 ;48 ; 65 ;53 ;39 ;36 ;33 ;27 ;23.

Pour le club B les âges des membres sont : 30 ; 24 ;33 ; 21 ;22 ; 35 ;27 ;28 ; 25 ;26 ;27 ; 42 ;40 ;35 ; 54 ; 29 ; 50 ; 30 ; 47 ; 31 ;33 ;34.

1. Entrer dans la liste **L1** les âges des membres du club A et dans la liste **L2** les âges des membres du club B.

2. Quel est l'âge moyen des membres du club A ? du club B ?

3. Déterminer la médiane et les quartiles pour chacune de ces séries statistiques.

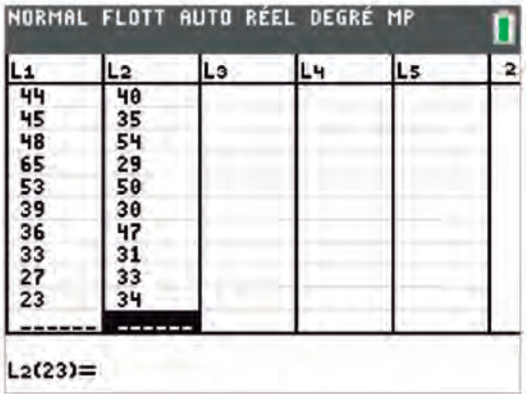
4. Représenter sur le même graphique les diagrammes en boîte des clubs A et B.

5. A l'aide de ces diagrammes en boîtes, peut-on affirmer que plus des ¾ des membres du club B sont plus jeunes que la moitié des membres du club A ?

## 1. Entrer les données

On commence par effacer les listes déjà présentes. Pour effacer toutes les listes en même temps on appuie sur [2nde] [mém +] **EffTtesListes**.

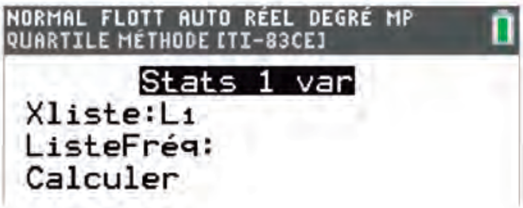
Puis dans [stats] **Modifier** on entre nos données.

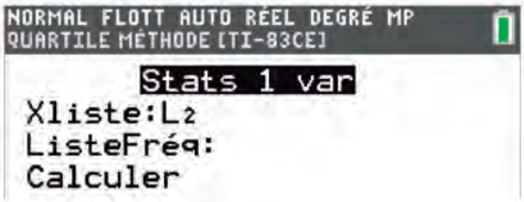



## 2. Calcul de l'âge moyen

Pour calculer les principaux indicateurs statistiques on appuie sur [stats] onglet **CALC** et choisir **Stats 1 Var**.

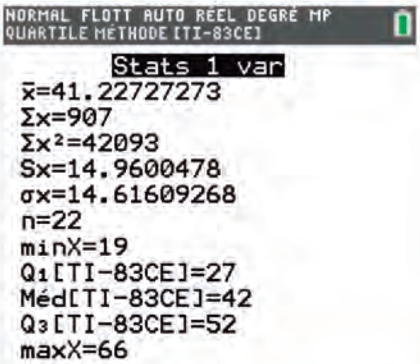
Il n'y a pas de fréquences ou d'effectifs associés aux valeurs du caractère on n'écrira rien dans **ListeFréq** (on peut aussi écrire **1**).



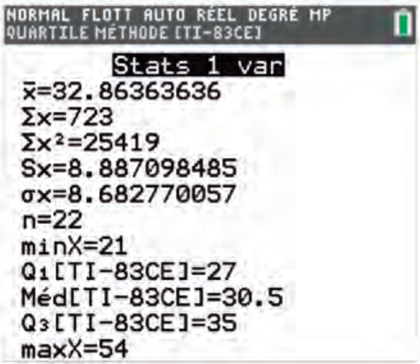



La moyenne d'âge des membres du club A est de $\overline{x_A} \approx 41{,}2$ et elle est de $\overline{x_B} \approx 32{,}9$ pour les membres du club B.

Club A




Club B




## 3. Indicateurs statistiques

A l'aide des résultats précédents on obtient :

| | 1er quartile $Q_1$ | Médiane $Me$ | 3ème quartile $Q_3$ |
|---|---|---|---|
| Club A | 27 | 42 | 52 |
| Club B | 27 | 30,5 | 35 |

## 4. Diagrammes en boîtes

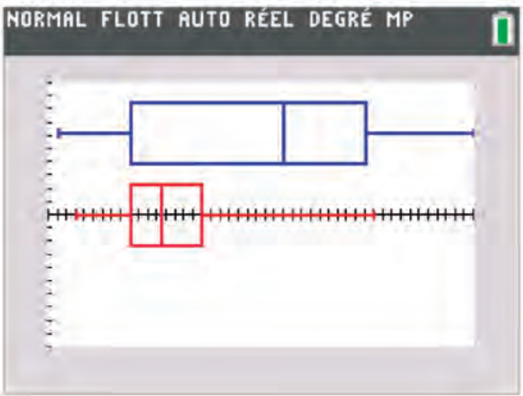
On peut représenter jusqu'à trois graphiques statistiques en même temps.

Pour le 1er diagramme en boîte : On appuie sur 2nde f(x) (graph stats f1) puis on sélectionne **Graph1**. On choisit le type diagramme en boîte. **Xliste** correspond aux valeurs du caractère qui sont dans **L1**, puis on choisit la couleur bleue.

On recommence avec **Graph2** pour les valeurs de la liste **L2**.

Pour ajuster automatiquement la fenêtre on choisit zoom **9:ZoomStat**.






## 5. L'affirmation est-elle vraie ?

Au moins les ¾ des membres du club B ont un âge compris entre 21 et 35 ans (entre la plus petite valeur et $Q_3$). La moitié des membres du club A ont 42 ans ou moins (médiane). L'affirmation est donc vraie.

# Effectifs cumulés et fréquences

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

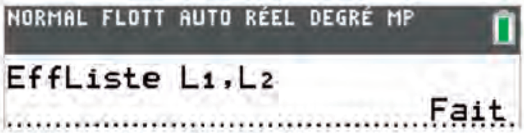
## Calculs autour des effectifs

Dans une usine qui fabrique des pièces, on a relevé les diamètres d'un lot.

| Diamètre en mm | 6,47 | 6,48 | 6,49 | 6,5 | 6,51 | 6,52 | 6,53 |
|---|---|---|---|---|---|---|---|
| Effectifs | 17 | 74 | 187 | 384 | 215 | 107 | 28 |

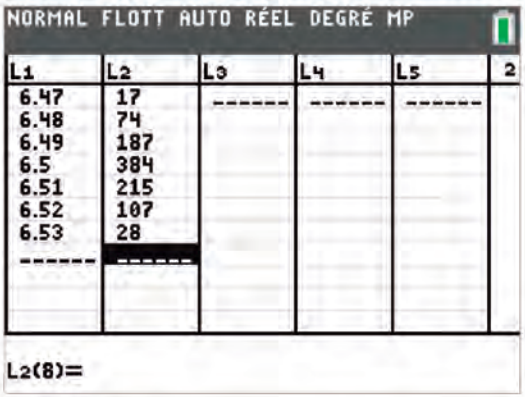
1. Entrer ces données dans votre calculatrice.
2. Calculer les fréquences avec 3 décimales de précision.
3. Calculer les fréquences cumulées croissantes.
4. En déduire le pourcentage de pièces du lot qui ont un diamètre inférieur ou égal à 6,49 puis celles qui ont un diamètre inférieure ou égale à 6,52.
5. Quel est le troisième quartile de cette série statistique ?

## 1. Entrer les données




Pour entrer les données on appuie sur [stats] `Modifier`. On efface éventuellement les listes présentes (voir une des activités précédentes) en appuyant sur [stats] `EffListe` (ici on efface les listes `L1` et `L2` avec la commande ci-contre.)

Remarque : `L1` et `L2` sont accessibles en appuyant sur [2nde] [1] et [2nde] [2] respectivement.

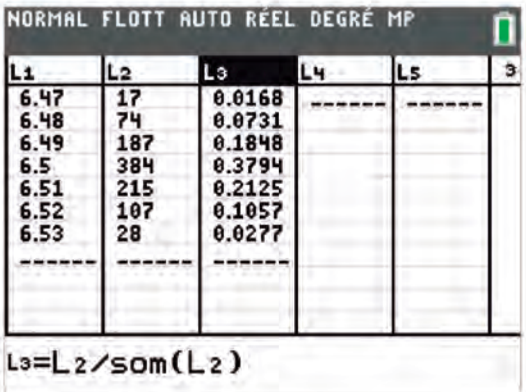



Puis on entre les données et on obtient les résultats ci-contre :

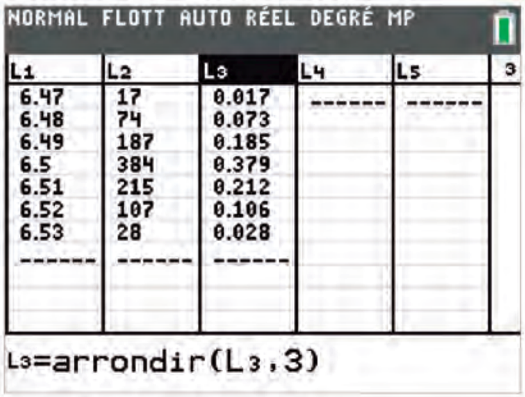
## 2. Calcul des fréquences

Pour calculer rapidement les fréquences, on se place (à l'aide des flèches de direction) tout en haut de la liste `L3` et on écrit `L2/som(L2)`.

L'instruction `som` est accessible en appuyant sur [2nde] [stats] onglet `MATH` et on choisit `som`.




On obtient ainsi toutes les fréquences associée (voir ci-dessous).



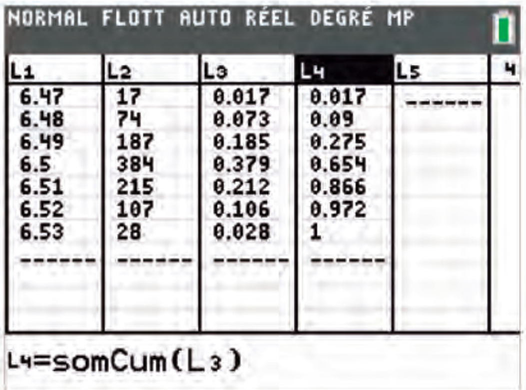
NORMAL FLOTT AUTO RÉEL DEGRÉ MP

| L1 | L2 | L3 | L4 | L5 |
|---|---|---|---|---|
| 6.47 | 17 | 0.017 | ------ | ------ |
| 6.48 | 74 | 0.073 | | |
| 6.49 | 187 | 0.185 | | |
| 6.5 | 384 | 0.379 | | |
| 6.51 | 215 | 0.212 | | |
| 6.52 | 107 | 0.106 | | |
| 6.53 | 28 | 0.028 | | |
| ------ | ------ | ------ | | |

L3=arrondir(L3,3)

Pour obtenir 3 décimales de précision on retourne en haut de la liste **L3**, on appuie sur [annul] pour effacer les instructions puis [math] onglet **NBRE** et on choisit **arrondir**. On complète l'instruction ainsi : **arrondir(L3,3)**.

## 3. Calcul des FCC



NORMAL FLOTT AUTO RÉEL DEGRÉ MP

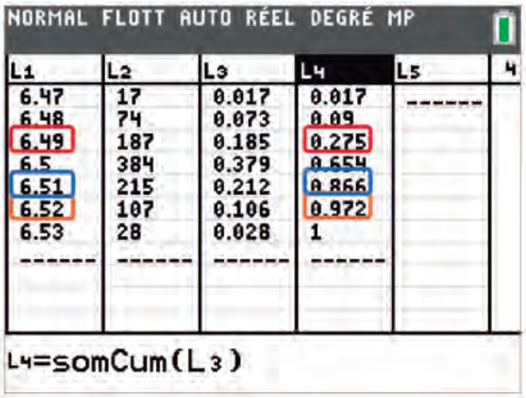
| L1 | L2 | L3 | L4 | L5 |
|---|---|---|---|---|
| 6.47 | 17 | 0.017 | 0.017 | ------ |
| 6.48 | 74 | 0.073 | 0.09 | |
| 6.49 | 187 | 0.185 | 0.275 | |
| 6.5 | 384 | 0.379 | 0.654 | |
| 6.51 | 215 | 0.212 | 0.866 | |
| 6.52 | 107 | 0.106 | 0.972 | |
| 6.53 | 28 | 0.028 | 1 | |
| ------ | ------ | ------ | ------ | |

L4=somCum(L3)

Pour calculer les fréquences cumulées croissantes, on se place tout en haut de **L4** et on appuie sur [2nde] [stats] (listes) onglet **OP** et on sélectionne **somCum** et on écrit **L3** entre parenthèses.

On obtient les résultats ci-contre.

## 4. Calcul de pourcentage



NORMAL FLOTT AUTO RÉEL DEGRÉ MP

| L1 | L2 | L3 | L4 | L5 |
|---|---|---|---|---|
| 6.47 | 17 | 0.017 | 0.017 | ------ |
| 6.48 | 74 | 0.073 | 0.09 | |
| 6.49 | 187 | 0.185 | 0.275 | |
| 6.5 | 384 | 0.379 | 0.654 | |
| 6.51 | 215 | 0.212 | 0.866 | |
| 6.52 | 107 | 0.106 | 0.972 | |
| 6.53 | 28 | 0.028 | 1 | |
| ------ | ------ | ------ | ------ | |

L4=somCum(L3)

Le pourcentage de pièces du lot qui ont un diamètre inférieur ou égal à 6,49 est 27,5%.

Le pourcentage de pièces du lot qui ont un diamètre inférieur ou égal à 6,52 est 97,2%.

## 5. Déterminer $Q_3$

Pour déterminer $Q_3$ il faut trouver la première fréquence cumulée croissante qui est supérieure ou égale à $0{,}75$. Ici c'est $0{,}866$.

$Q_3$ est la valeur du caractère associé. Ici on trouve $Q_3 = 6{,}51$.

# S'approprier l'environnement Python

TI-82 Advanced Edition Python

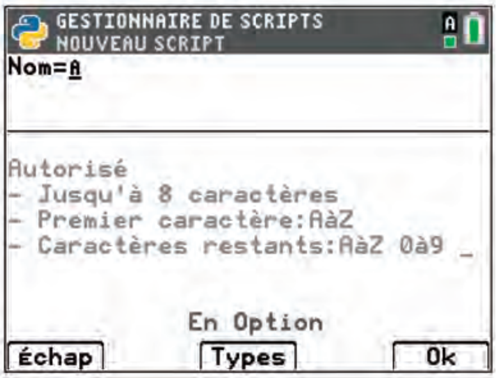
TI-83 Premium CE Edition Python

## Contexte

La calculatrice permet de développer en Python.

On va voir comment mettre en œuvre ce langage de programmation avec la construction d'une première fonction **CM** qui transforme un pourcentage passé en paramètre, en son équivalent sous forme de coefficient multiplicateur. Si l'on souhaite augmenter un prix de 30%, la fonction renverra le coefficient multiplicateur 1,3.

## 1. Créer un nouveau script

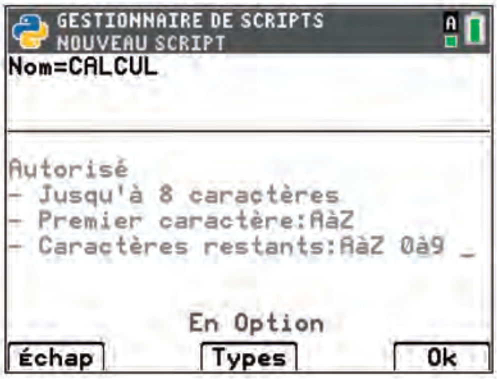



On accède à l'éditeur Python à l'aide de la touche prgm puis en exécutant l'application Python.

Pour créer un nouveau script, on sélectionne l'onglet Nouv à l'aide de la touche zoom qui est située sous l'onglet visé.

On obtient alors la fenêtre ci-contre. Le curseur nous indique que l'on peut saisir du texte en continu. Si on veut saisir un chiffre, je dois alors désactiver la saisie de caractères en appuyant sur la combinaison de touches 2nde + alpha

Le curseur change d'aspect et la lettre A disparait



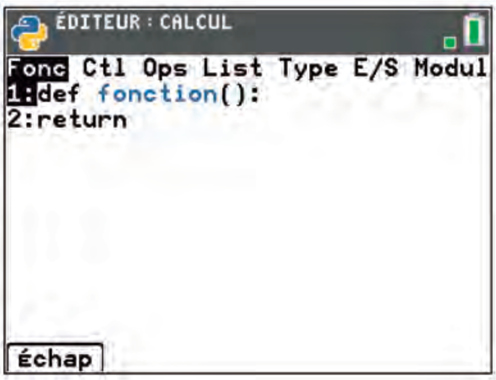

On nomme notre script **CALCUL**. On pourrait se servir de l'onglet Types pour définir le type de contenu à venir dans notre script. Cela facilite notamment la saisie de certains imports de librairie (`random`, `math` par exemple) mais n'est pas obligatoire. On aura la possibilité de les importer par la suite.

On valide la création du script à l'aide de l'onglet Ok avec la touche graphe

## 2. Saisir une nouvelle fonction

On va créer une première fonction **CM** qui prend en paramètre **p** un taux d'évolution et qui renvoie le coefficient multiplicateur correspondant.

La calculatrice offre une interface de saisie efficace que nous allons découvrir.



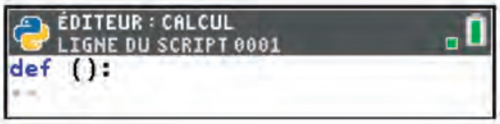

Commençons par aller dans l'onglet Fns… à l'aide de la touche f(x)

Cela nous permet d'accéder à des menus contenant la plupart des fonctions dont nous aurons besoin. On peut ainsi les repérer en naviguant dans les onglets à l'aide des flèches de la calculatrice et les sélectionner à l'aide de la touche entrer

Une fois validées, les instructions, structures, fonctions etc. sont importées dans notre script sans que nous ayons besoin de les saisir. C'est un gain de temps appréciable.

On valide donc l'instruction **def fonction() :**

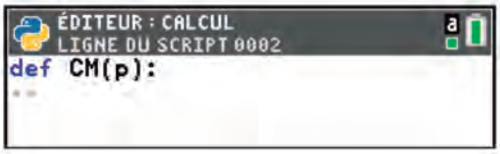
Notre script est implémenté par ces premiers éléments.




Il faut activer le mode « écriture continue » (« n'écrire que des lettres ») à l'aide des touches 2nde alpha pour écrire en minuscule

ou 2nde alpha alpha pour écrire en majuscule

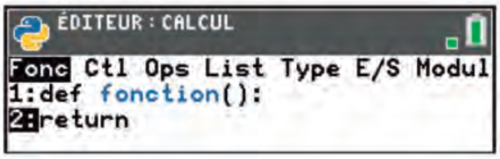
Une fois activé, pour passer d'un mode à l'autre, il suffit d'appuyer seulement sur alpha




On complète notre première ligne de script de la manière ci-contre.

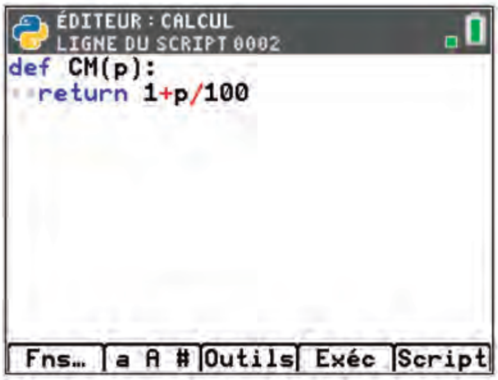
Les points gris sont des marqueurs d'indentation. Nous aurons l'occasion d'en reparler.

Si on les efface par mégarde, il est possible de corriger une ligne à l'aide de l'onglet Outils via la touche zoom .



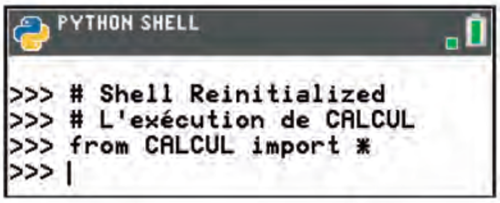

On y trouvera les commandes pour augmenter ou diminuer l'indentation.

On complète la deuxième ligne du script en allant chercher l'instruction return dans l'onglet Fns…



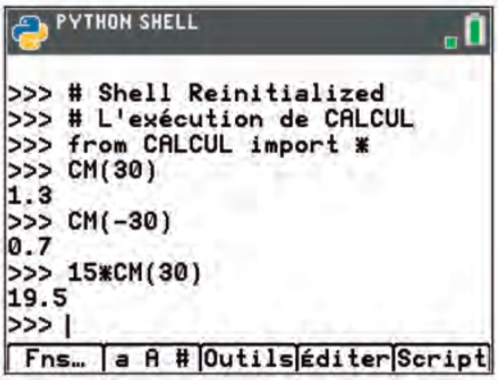

On quitte le mode « écriture continue » pour saisir nos différentes valeurs et opérateurs. Ponctuellement on active la saisie d'un caractère en appuyant une seule fois sur la touche alpha

Notre saisie terminée, on va pouvoir tester notre première fonction dans la console Python en se rendant dans l'onglet Exéc à l'aide de la touche trace




Un message sur l'exécution du script apparait.

On peut saisir en toutes lettres notre instruction ou appuyer sur la touche var pour sélectionner le nom de la fonction et compléter la valeur passée en paramètre.




Testons les retours de `CM(30)` et `CM(-30)`

La fonction nous renvoie bien les coefficients multiplicateurs 1,3 et 0,7

Augmentons 15 de 30% en saisissant `15*CM(30)`

On obtient bien 19,5

# S'approprier l'environnement Python

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Contexte

On poursuit notre travail d'appropriation en créant un nouveau script à partir du précédent.

L'objectif est la construction d'une fonction `Capital` qui calcule, pour un placement initial, le montant récupéré au bout de t années à un certain pourcentage. Toutes ces informations devront être fournies en paramètres à la fonction. Si on place 1000 euros pendant 5 ans à 1%, l'instruction `Capital(1000,1,5)` renverra 1010.

```
GESTIONNAIRE DE SCRIPTS
GÉRER
App Python : v5.5.2.0044

1:Dupliquer le script…
2:Supprimer le script…    [suppr]
3:Renommer le script…
4:A propos de…
5:Quitter Python

Échap
```

## 1. Dupliquer un script

Lorsqu'on lance l'environnement Python, on arrive en premier dans le gestionnaire de script. En positionnant le curseur devant le nom d'un script existant, il est possible de l'éditer à l'aide de l'onglet [Édit] mais également de le dupliquer pour en recopier le contenu dans un nouveau.

Pour cela, il faut se rendre dans l'onglet [Gérer] à l'aide de la touche [graphe]

On accède à plusieurs options pour gérer nos scripts.

On a choisi de dupliquer le script **CALCUL** précédent pour appeler le nouveau **EPARGNE**

```
GESTIONNAIRE DE SCRIPTS
DUPLIQUER LE SCRIPT
Nom actuel : CALCUL
Nom=EPARGNE
```

## 2. Ajouter une nouvelle fonction

Nous allons créer une nouvelle fonction `AppliqPourcent` dans notre script.

Elle prend en paramètre un montant **m** et un pourcentage **p**. Elle renvoie un nouveau montant auquel on aura appliqué l'évolution **p**.

Lorsque nous positionnons le curseur en fin de ligne de la fonction `CM` et validons, l'éditeur Python choisit de maintenir l'indentation précedente.

Il faut la supprimer pour pouvoir créer une nouvelle fonction, à l'aide de la touche [suppr]

On utilise la fonction `CM` pour convertir le pourcentage en coefficient multiplicateur.

Il est important de se souvenir que le langage Python est également sensible à la « casse ». C'est-à-dire qu'il faut respecter les majuscules et les minuscules lors de la saisie de nos fonctions.

Dans l'exemple ci-contre, la saisie dans la console de l'instruction `CM(30)` fonctionne tandis que `cm(30)` renvoie une erreur.

```
ÉDITEUR : EPARGNE
LIGNE DU SCRIPT 0005
def CM(p):
  return 1+p/100

def AppliqPourcent(m,p):
  return m*CM(p)

Fns… | a A # | Outils | Exéc | Script
```

```
PYTHON SHELL
>>> from EPARGNE import *
>>> CM(30)
1.3
>>> cm(30)
Traceback (most recent call last
):
  File "<stdin>", line 1, in <mo
dule>
NameError: name 'cm' is not defi
ned
>>> |
Fns… | a A # | Outils | Éditer | Script
```

On peut vérifier en console que si on applique une augmentation de 30% à 15 euros, on obtient bien 19,50 euros

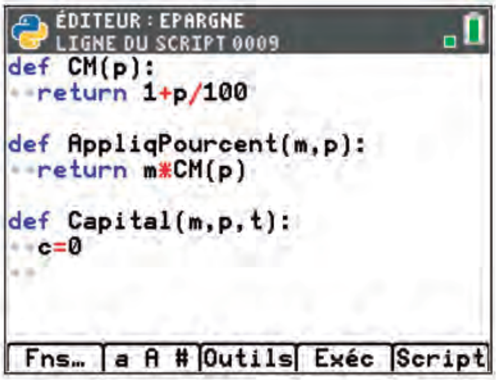
```
PYTHON SHELL

>>> # Shell Reinitialized
>>> # L'exécution de EPARGNE
>>> from EPARGNE import *
>>> AppliqPourcent(15,30)
19.5
```

# S'approprier l'environnement Python



## 3. Connaitre des raccourcis



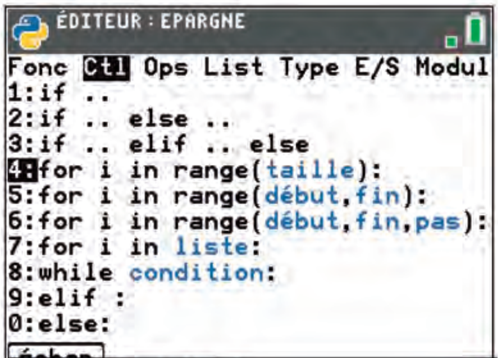

Pour accélérer encore davantage la saisie de script Python, il est intéressant de connaitre certaines manipulations que nous allons découvrir lors de la saisie de notre fonction `Capital`

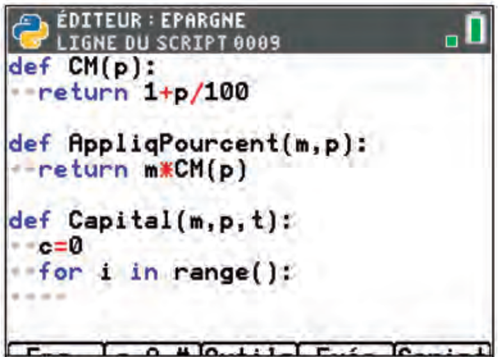
Par exemple, la calculatrice offre plusieurs raccourcis clavier.

Le signe = est accessible directement en appuyant sur la touche sto→

Comme pour la définition de fonction, il est possible d'importer des structures de contrôle que l'éditeur de la calculatrice se chargera d'indenter.

Il ne reste alors, ici, plus qu'à compléter l'instruction `range()` avec la variable `t`.

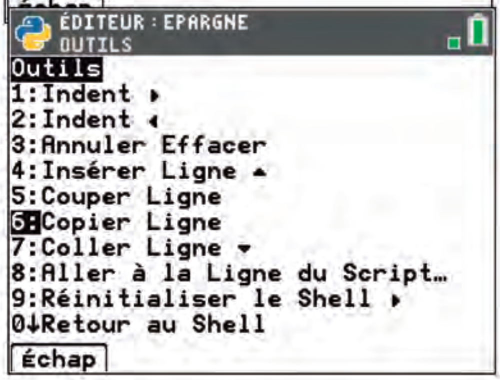


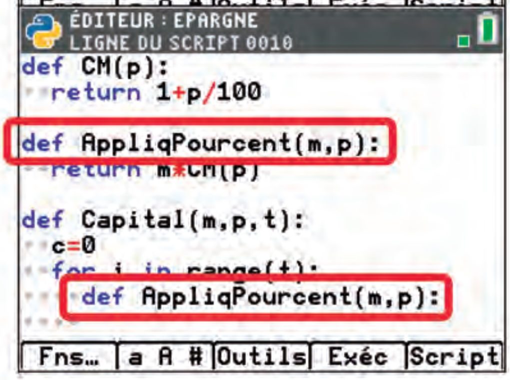




Nous souhaitons saisir l'instruction :
`c=AppliqPourcent(m,p)`

L'écriture peut être fastidieuse mais là encore, la calculatrice dispose de fonctionnalité pour accélérer la saisie comme un copier/coller disponible dans l'onglet `Outils`

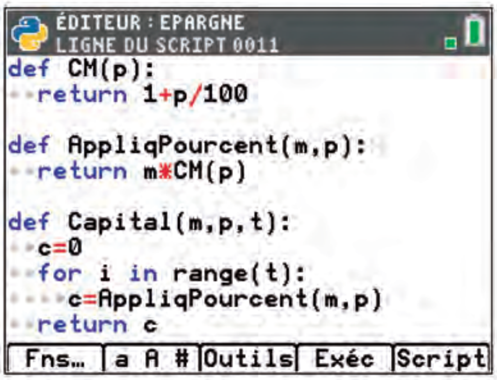
Il ne reste plus que quelques modifications à faire manuellement.




Pour conclure, voici le script complet que vous devez obtenir.

Vous pourrez tester en console que 1000 euros placés pendant 5 année à 1% d'intérêts par an rapporteront 10 euros puisque le capital sera alors de 1010 euros.

# Diviseurs d'un entier



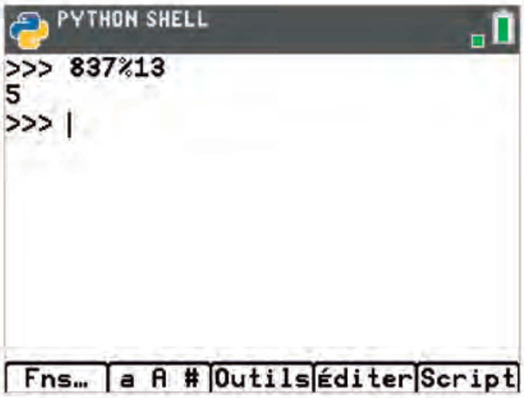
## Diviseur d'un entier non nul

**Définition** : Soit $a$ et $b$ deux entiers non nuls. On dit que $b$ divise $a$ dès qu'il existe $k \in \mathbb{Z}$ tel que $a = kb$.

**Remarque** : $b$ divise $a$ signifie que lorsqu'on effectue la division euclidienne de $a$ par $b$ alors le reste est nul.

En Python pour obtenir le reste de la division euclidienne de $a$ par $b$ on écrit **a%b**.

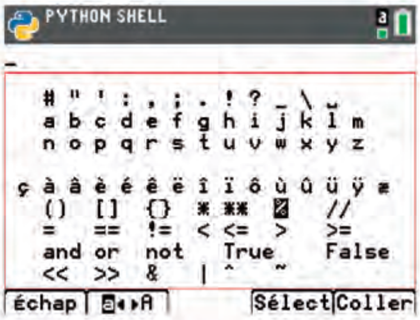
Accédons à la programmation en Python en appuyant sur [prgm] puis **Python App**. Commençons en exécutant quelques commandes dans le Shell (la console) en sélectionnant [**Shell**] :




Exemple 1 : Déterminer si 13 divise 837, en calculant le reste de la division euclidienne de 837 par 13.

Pour cela on va écrire : **837%13**.

Pour obtenir le symbole %, dans [**a A #**], à l'aide des flèches de direction, sélectionner le symbole % puis appuyer sur **Sélect** et **Coller** afin de l'insérer dans le Shell.




Le reste de la division euclidienne de 837 par 13 est 5, donc 13 ne divise pas 837.

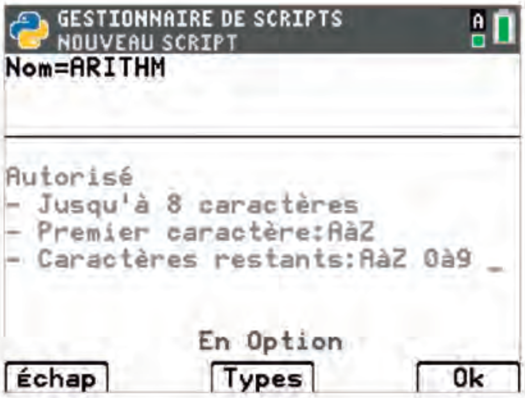
Exemple 2 : Déterminer si 17 divise 493.


On entre dans la console 493%17 est on obtient 0. Ce qui prouve que 17 divise 493.

Ecrivons maintenant une fonction **divise** qui prend comme arguments deux entiers non nuls **a** et **b** et qui renvoie **True** si **a** divise **b** et **False** sinon.

**Application** : Utiliser cette fonction avec les deux exemples précédents.

Créer un nouveau script en appuyant sur [**Script**] (pour entrer dans le gestionnaire de scripts) puis [**Nouv**]. Ecrire le nom du script (par exemple **ARITHM**) et valider en appuyant sur [**Ok**].



# Diviseurs d'un entier



`def` et `return` sont accessibles dans [Fns…].

Le test conditionnel `if` … `else` est accessible dans [Fns…] onglet `Ctl`.

`True` et `False` peuvent être saisies au clavier ou accessibles dans [Fns…] onglet `Ops` (tout en bas de la liste).

Exécuter le script en appuyant sur [Exéc], on arrive alors dans le Shell. Pour obtenir rapidement le nom de notre fonction on appuie sur [var]

On retrouve bien les résultat précédents (voir copie d'écran ci-contre).

```
ÉDITEUR : ARITHM
LIGNE DU SCRIPT 0005
def divise(a,b):
  if b%a==0:
    return True
  else:
    return False
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de ARITHM
>>> from ARITHM import *
>>> divise(13,837)
False
>>> divise(17,493)
True
```

## Nombres parfaits

**Définition** : Soit $a$ un entier non nul. On dit que $a$ est un nombre parfait si la somme de ses diviseurs stricts est égale à $a$.

Remarque : On ne prend que les diviseurs positifs de $a$. Un diviseur strict de $a$ est un diviseur de $a$ qui est différent de $a$.

Exemple 3 : 6 est-il parfait ? 10 est-il parfait ?

Les diviseurs stricts de 6 sont 1 ; 2 ; 3. De plus $1+2+3=6$. Donc 6 est un nombre parfait.

Les diviseurs stricts de 10 sont 1 ; 2 ; 5. De plus $1+2+5=8$. Donc 10 n'est pas un nombre parfait.

Ecrivons une fonction Python `parfait` qui prend comme argument un entier non nul **a** et qui renvoie `True` si **a** est parfait et `False` sinon.

Pour cela on va écrire une boucle avec un entier **b** qui va varier de **1** à `a-1` (on s'arrête à `a-1` car on ne cherche que des diviseurs stricts de **a**).

De plus on va utiliser une variable **s** qui va représenter la somme des diviseurs stricts de **a**. Ainsi dans la boucle, on va tester si **b** divise **a**, et si c'est le cas on ajoutera ce diviseur à `s`.

On obtient le script ci-contre :

```
ÉDITEUR : ARITHM
LIGNE DU SCRIPT 0016
def parfait(a):
  s=0
  for b in range(1,a):
    if divise(b,a):
      s=s+b
  if s==a:
    return True
  else:
    return False
Fns… | a A # | Outils | Exéc | Script
```

**Application** : Parmi les nombres suivants, déterminer ceux qui sont parfaits : 28 ;136 ;496.

A l'aide de notre fonction `parfait` on peut conclure :

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de ARITHM
>>> from ARITHM import *
>>> parfait(28)
True
>>> parfait(136)
False
>>> parfait(496)
True
>>> |
Fns… | a A # | Outils | Éditer | Script
```

**Conclusion** : 28 est parfait, 136 n'est pas parfait et 496 est parfait.

# Calculs de déterminants

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Déterminant de deux vecteurs

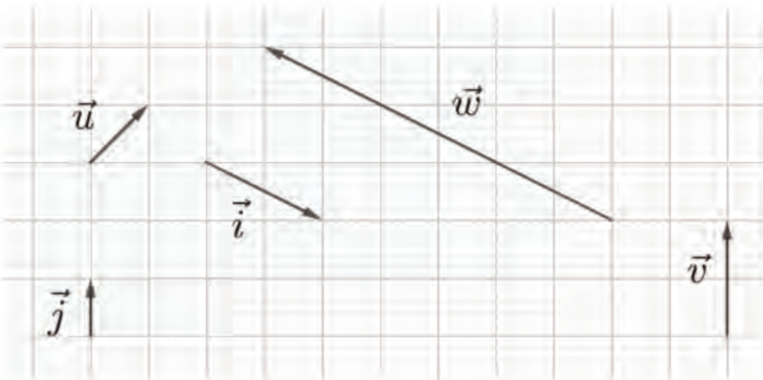
Le déterminant de deux vecteurs permet de déterminer si les vecteurs sont colinéaires (c'est-à-dire s'ils ont la même direction).

**Exemple** : En s'aidant de la figure ci-contre, déterminer les vecteurs qui sont colinéaires :

**Solution** : Les vecteurs $\vec{\imath}$ et $\vec{w}$ sont colinéaires.

Les vecteurs $\vec{\jmath}$ et $\vec{v}$ sont aussi colinéaires.

Le vecteur $\vec{u}$ n'est colinéaire à aucun des vecteurs représentés.




**Définition** : Soit $(\vec{\imath},\vec{\jmath})$ une base de l'ensemble des vecteurs du plan. Soit $\vec{u}\begin{pmatrix}x\\y\end{pmatrix}$ et $\vec{v}\begin{pmatrix}x'\\y'\end{pmatrix}$ dans la base $(\vec{\imath},\vec{\jmath})$, avec $x, y, x'$ et $y'$ des réels.

Le déterminant des vecteurs $\vec{u}$ et $\vec{v}$ se note :

$\det(\vec{u},\vec{v})$ ou encore $\begin{vmatrix}x & x'\\ y & y'\end{vmatrix}$ et a pour valeur :

$\det(\vec{u},\vec{v}) = xy' - x'y$

**Propriété** : $\vec{u}$ et $\vec{v}$ sont colinéaires si et seulement si $\det(\vec{u},\vec{v}) = 0$

<u>Exemple 1</u> : On donne $\vec{u}\begin{pmatrix}\frac{3}{4}\\\frac{3}{2}\end{pmatrix}$ et $\vec{v}\begin{pmatrix}\frac{6}{7}\\\frac{5}{7}\end{pmatrix}$ dans la base $(\vec{\imath},\vec{\jmath})$. Les vecteurs $\vec{u}$ et $\vec{v}$ sont-ils colinéaires ?

Calculons le déterminant des vecteurs $\vec{u}$ et $\vec{v}$ :

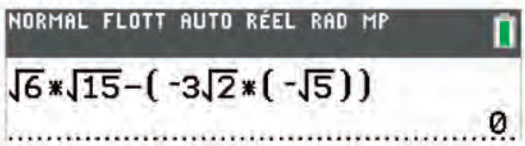
$$\det(\vec{u},\vec{v}) = \begin{vmatrix}\frac{3}{4} & \frac{6}{7}\\ \frac{3}{2} & \frac{5}{7}\end{vmatrix} = \frac{3}{4}\times\frac{5}{7} - \frac{6}{7}\times\frac{3}{2} = \frac{15}{28} - \frac{9}{7} = \frac{15}{28} - \frac{36}{28} = -\frac{21}{28} = -\frac{3}{4}$$

On peut s'aider de sa calculatrice pour vérifier ses calculs et vérifier aussi que la fraction obtenue est irréductible (c'est-à-dire qu'on ne peut pas plus la simplifier).
Ainsi $\det(\vec{u},\vec{v}) \neq 0$ les vecteurs $\vec{u}$ et $\vec{v}$ ne sont pas colinéaires.


<u>Exemple 2</u> : On donne $\vec{u}\begin{pmatrix}\sqrt{6}\\-\sqrt{5}\end{pmatrix}$ et $\vec{v}\begin{pmatrix}-3\sqrt{2}\\\sqrt{15}\end{pmatrix}$ dans la base $(\vec{\imath},\vec{\jmath})$. Les vecteurs $\vec{u}$ et $\vec{v}$ sont-ils colinéaires ?

Calculons $\det(\vec{u},\vec{v}) = \begin{vmatrix}\sqrt{6} & -3\sqrt{2}\\ -\sqrt{5} & \sqrt{15}\end{vmatrix} = \sqrt{6}\times\sqrt{15} - \left(-3\sqrt{2}\times(-\sqrt{5})\right)$

$$= \sqrt{90} - 3\sqrt{10} = \sqrt{9\times 10} - 3\sqrt{10}$$
$$= \sqrt{9}\times\sqrt{10} - 3\sqrt{10} = 3\sqrt{10} - 3\sqrt{10}$$




Donc $\det(\vec{u},\vec{v}) = 0$ les vecteurs $\vec{u}$ et $\vec{v}$ sont colinéaires.

# Calculs de déterminants

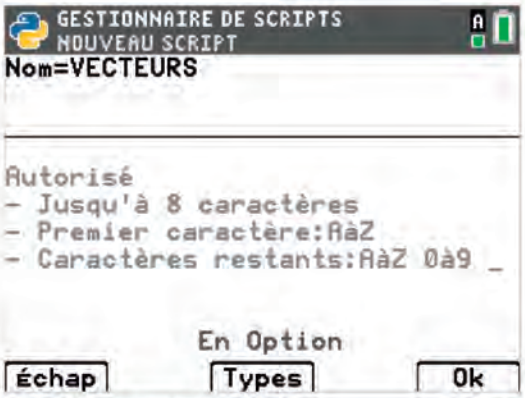
TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

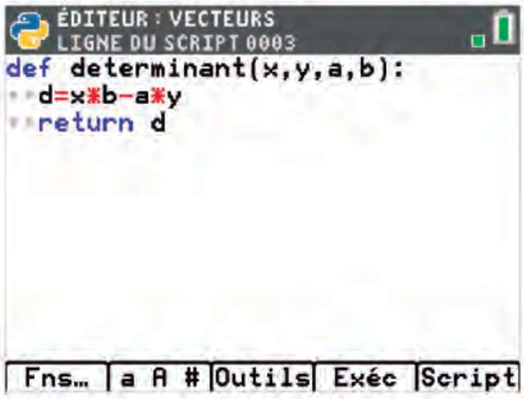
## Script Python pour calculer le déterminant

Il est possible d'automatiser le calcul du déterminant à l'aide d'un script simple en Python :

```
def determinant(x,y,a,b) :
••d=x*b-a*y
••return d
```




Nommons notre script **VECTEURS** et entrons le script précédent :

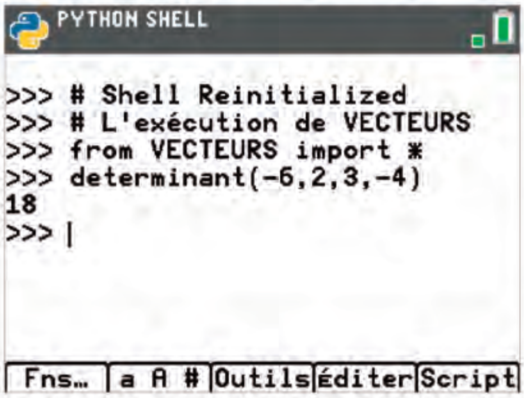



Utilisons maintenant notre fonction **determinant** dans l'exemple suivant :

Exemple 3 : Calculer le déterminant des vecteurs $\vec{u}\begin{pmatrix}-6\\2\end{pmatrix}$ et $\vec{v}\begin{pmatrix}3\\-4\end{pmatrix}$ dont les coordonnées sont données dans une base $(\vec{\imath},\vec{\jmath})$.

On commence par exécuter notre script en appuyant sur **Exéc** puis, dans la console, on va appeler notre fonction **determinant** qui est maintenant en mémoire.

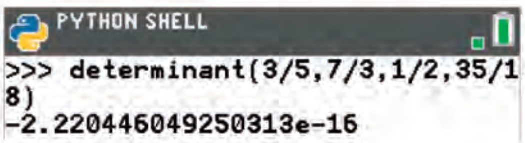
Dans la console, pour accéder rapidement aux fonctions de notre script on appuie sur [var] puis on entre les arguments de la fonction **determinant**.




Conclusion : $\det(\vec{u},\vec{v}) \neq 0$.

Exemple 4 :A l'aide de notre script Python précédent, calculer le déterminant des vecteurs $\vec{u}\begin{pmatrix}\frac{3}{5}\\\frac{7}{3}\end{pmatrix}$ et $\vec{v}\begin{pmatrix}\frac{1}{2}\\\frac{35}{18}\end{pmatrix}$ dont les coordonnées sont données dans une base $(\vec{\imath},\vec{\jmath})$, puis déterminer s'ils sont colinéaires.

Dans la console on écrit (voir copie d'écran ci-contre) :




**Attention** ! Lorsqu'on ne travaille pas avec des nombres entiers, Python utilise un mode de calcul approximatif. Si le résultat d'un calcul est 0, il se peut que le résultat affiché soit un nombre excessivement proche de 0 (mais pas 0…). Il faudra donc l'interpréter comme 0 (ce qu'on obtiendrait avec un calcul « à la main »).

Ainsi on peut affirmer que $\det(\vec{u},\vec{v}) = 0$ donc les vecteurs $\vec{u}$ et $\vec{v}$ sont colinéaires.

# Système d'équations



## Deux équations et deux inconnues

Exemple 1 : Une entreprise invite ses 115 employés lors d'un repas de fin d'année. Il y avait deux menus au choix :

La formule 1 à 17€ et la formule 2 à 23€.

L'addition étant de 2237€ combien d'employés ont choisi la formule 1 ?

Pour résoudre ce type de problème, on commence par donner un nom aux inconnues. On notera :

- $x$ le nombre d'employés ayant pris la formule 1
- $y$ le nombre d'employés qui ont pris la formule 2.

D'après l'énoncé, il y a 115 employés donc $x + y = 115$.

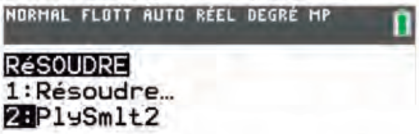
D'autre part il y a $x$ personnes qui ont pris le menu à 17€ et $y$ personnes qui ont pris le menu à 23€. Cela représente une addition de $17x + 23y$, or on sait que ce montant est 2237€, ainsi $17x + 23y = 2237$.

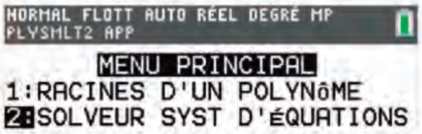
Il nous faut maintenant résoudre le système d'équations ci-contre :

$$\begin{cases} x + y = 115 \\ 17x + 23y = 2237 \end{cases}$$

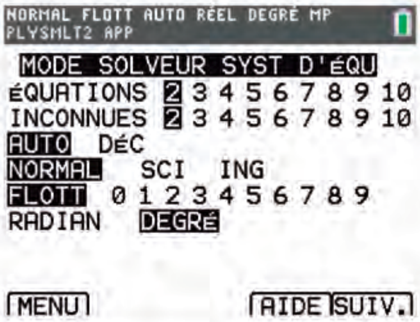
Pour résoudre ce système à l'aide de notre calculatrice, on appuie sur résol

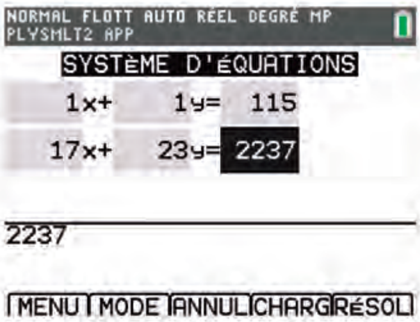
On choisit **PlySmlt2** puis **SOLVEUR SYST D'ÉQUATIONS** :

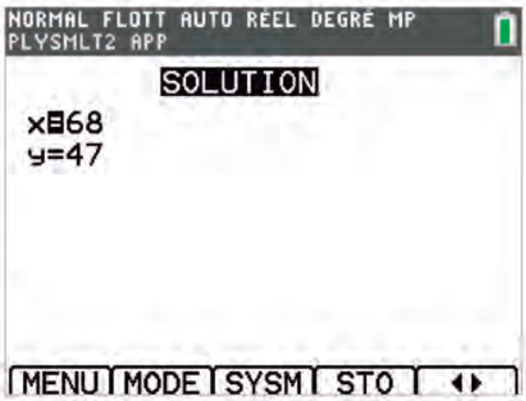



On choisit 2 équations et 2 inconnues puis on entre les coefficients et enfin on appuie sur **RÉSOL** (touche graphe) pour obtenir les solutions :




On obtient $\begin{cases} x = 68 \\ y = 47 \end{cases}$. Il y a eu 68 formules 1 et 47 formules 2.

Pour sortir de cette application appuyer sur 2nde mode (quitter).

# Système d'équations

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Intersection de deux droites

Exemple 1 : Dans un repère $(O, \vec{\imath}, \vec{\jmath})$ on donne les droites d'équation :

$D_1 : 3x + 2y - 1 = 0$ ; $D_2 : 5x - y - 3 = 0$. On admet que $D_1$ et $D_2$ sont sécantes.

Déterminer les coordonnées du point d'intersection des droites $D_1$ et $D_2$.

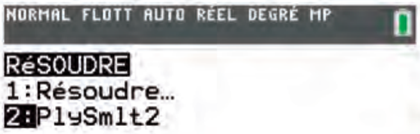
Pour déterminer les coordonnées d'un éventuel point d'intersection entre les droites $D_1$ et $D_2$ on va commencer par résoudre le système :

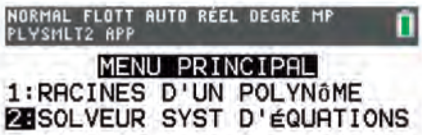
$$\begin{cases} 3x + 2y - 1 = 0 \\ 5x - y - 3 = 0 \end{cases}$$

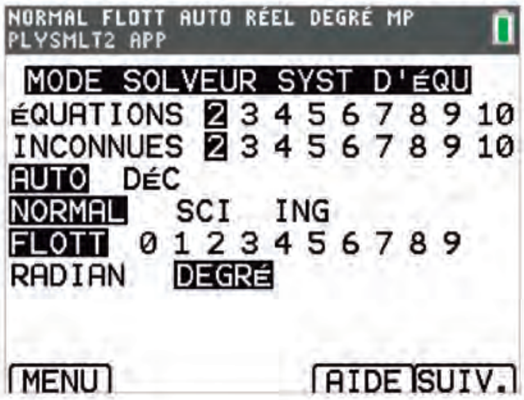
Pour utiliser notre solveur de systèmes d'équations, il faut légèrement adapter notre système précédent :

$$\begin{cases} 3x + 2y = 1 \\ 5x - y = 3 \end{cases}$$

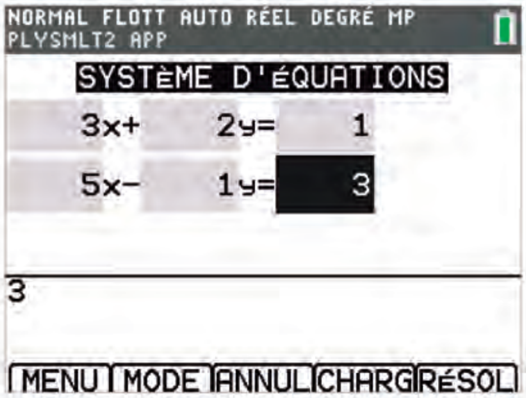
Appuyons donc sur [résol] puis **PlySmlt2** et **SOLVEUR SYST D'ÉQUATIONS** :

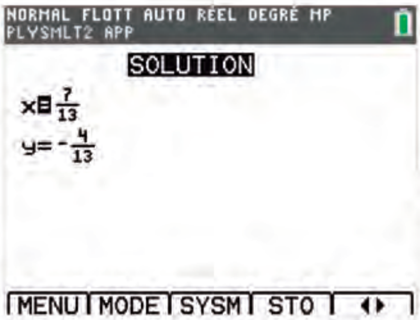


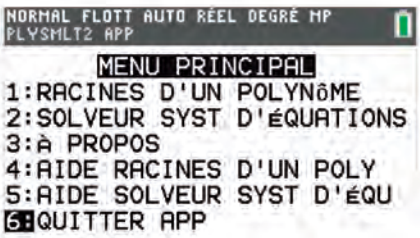




On sélectionne 2 équations et 2 inconnues, on entre les coefficients et enfin on appuie sur **RÉSOL** (touche [graphe]) pour obtenir les solutions :




Remarque : Pour sortir de l'application **PlySmlt2** on appuie sur [2nde] [mode] (quitter).

Conclusion : En notant $A$ le point d'intersection des droites $D_1$ et $D_2$, on trouve : $A\left(\frac{7}{13}; -\frac{4}{13}\right)$ dans le repère $(O, \vec{\imath}, \vec{\jmath})$.

# Gérer des variables dans des formules

TI-82 Advanced Edition Python

TI-83 Premium CE Edition Python

## Contexte

Il peut être utile de stocker des nombres, longs à saisir, dans ce que l'on appelle des variables pour pouvoir s'en resservir à plusieurs reprises, sans avoir à les ressaisir. De même, on peut être amené à répéter plusieurs calculs identiques où seules certaines données changent et là encore, il sera utile d'avoir recours à l'utilisation de variables.

Etudions le volume d'un cône de révolution de hauteur $\frac{\sqrt{3}}{2}$ cm et de rayon $\frac{\sqrt{5}+1}{2}$ cm. Encadrons ce même volume lorsque le rayon varie entre 3cm et 5 cm.

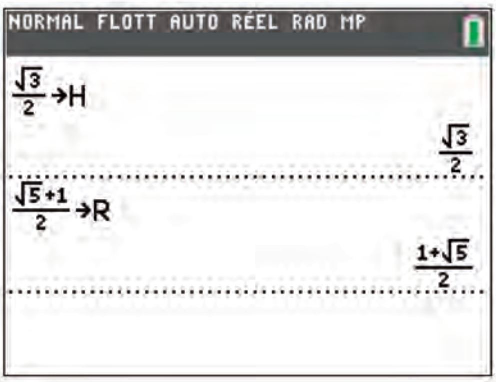
Si l'énoncé peut sembler effrayant, la calculatrice va nous accompagner utilement dans ces calculs ;-)

## 1. Stocker une valeur dans une variable

Les noms de variables sont prédéfinis : **A**, **B**, **C** …

On va enregistrer (stocker) la hauteur dans la variable **H** et le rayon dans la variable **R**.

Pour cela, on saisit la valeur $\frac{\sqrt{3}}{2}$ à l'aide des touches 2nde + $x^2$ pour faire apparaitre la racine carrée puis $\frac{□}{□}$ pour saisir le dénominateur de la fraction.




La flèche, qui va nous servir à indiquer dans quelle variable stocker notre valeur, est obtenue à l'aide de la touche sto→ .

Enfin, on indique le nom de la variable retenue à l'aide de la combinaison de touches alpha + **la lettre souhaitée**.

Par exemple : alpha + ^ pour **H** et alpha + × pour **R**.

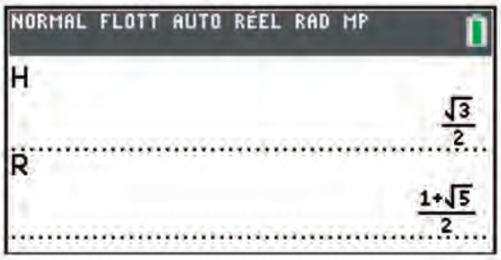
On valide avec la touche entrer

## 2. Utiliser la valeur d'une variable

Pour utiliser le contenu de la variable, il suffit désormais de saisir dans le calcul souhaité le nom de la variable à l'aide de la combinaison de touches alpha + **la lettre de la variable**

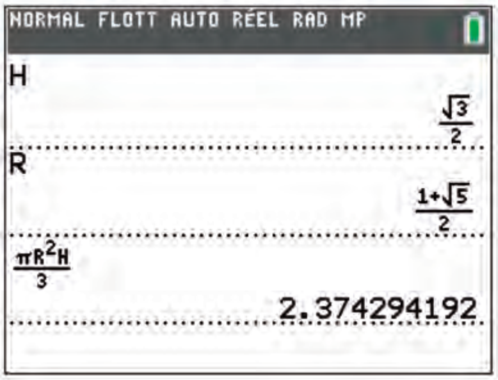
alpha + ^ + entrer

alpha + × + entrer



# Gérer des variables dans des formules






Ainsi on peut maintenant calculer le volume à l'aide de la formule $V_{cône} = \frac{\pi R^2 h}{3}$

$\pi$ s'obtient à l'aide des touches 2nde + trig

On peut élever **R** au carré à l'aide de la touche x²

Le volume de notre cône vaut approximativement $2{,}4\ cm^3$

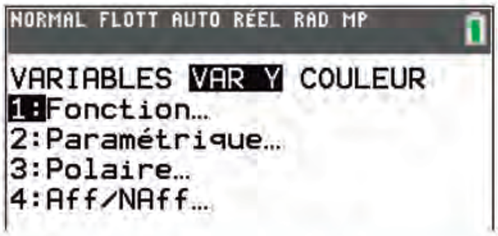

Changeons maintenant la valeur contenue dans la variable **R** par 3 et recalculons le nouveau volume.

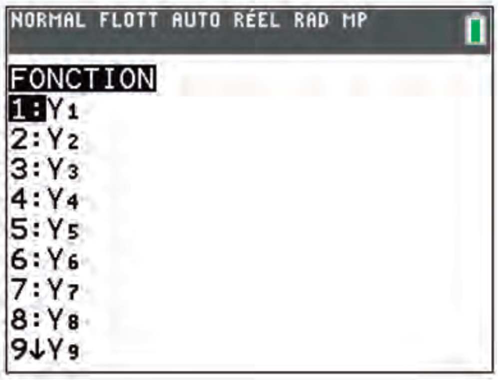
On recommence en plaçant la valeur 5 dans la variable **R** et recalculons le nouveau volume.

Le volume varie approximativement entre $8\ cm^3$ et $22{,}7\ cm^3$

Pour rappeler ces différentes formules, pas besoin de les ressaisir, vous pouvez naviguer dans l'historique de la calculatrice qui s'active automatiquement lorsque vous appuyez sur les touches ^ ou ˅.

## 3. La touche « var »

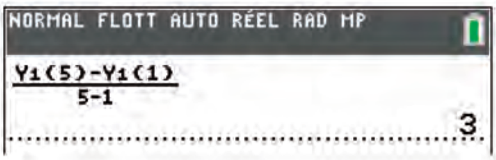



Vous aurez, sans doute, remarqué la touche var de votre calculatrice.

Elle permet d'accéder à un certain nombre de variables prédéfinies comme par exemple, les noms de fonctions saisies dans l'éditeur de fonctions dans l'onglet **VAR Y** (utilisé notamment avec la fiche « 8 - Vérification d'un tableau de signes »).

Ainsi, on peut vérifier, par le calcul, que le taux d'accroissement entre 1 et 5 de la fonction affine d'expression $3x + 5$ et saisie dans l'éditeur de fonctions en **Y1** vaut bien son coefficient directeur 3.



# Importer des librairies
# Utiliser les listes

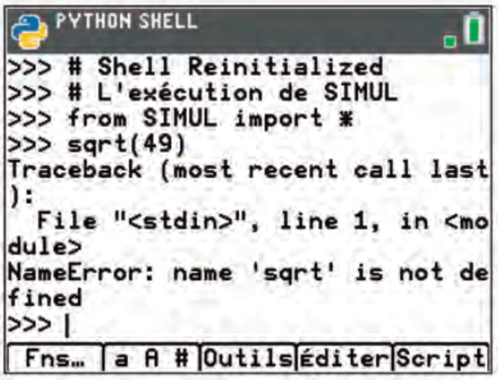
TI-82 Advanced Edition Python
TI-83 Premium CE Edition Python

## Contexte

On va utiliser l'environnement **Python** pour réaliser la simulation de 100 lancers d'un dé à six faces.

On va s'intéresser aux nombres de 6 obtenus à la fin de nos lancers.

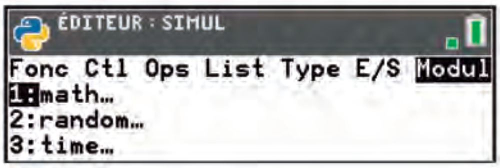
## 1. Importer des librairies




On crée un script `SIMUL` vierge et on l'exécute immédiatement en appuyant sur [Exéc]

On appuie sur les touches [2nde] + [x²]. Cela fait apparaitre l'instruction `sqrt()` que l'on complète avec le paramètre `49`, pour demander à obtenir $\sqrt{49}$. On obtient un message d'erreur.

La fonction n'est pas une fonction incluse par défaut en **Python**. Elle fait partie d'une librairie, c'est-à-dire une sorte de module comportant un certain nombre de fonctions supplémentaires. `Sqrt()` fait partie de la librairie `math`.




On retourne dans notre script ([Éditer]).

Puis [Fns…] et enfin onglet `Modul`. On commence par se rendre dans l'item `1:math…` et on sélectionne `1:from math import *`

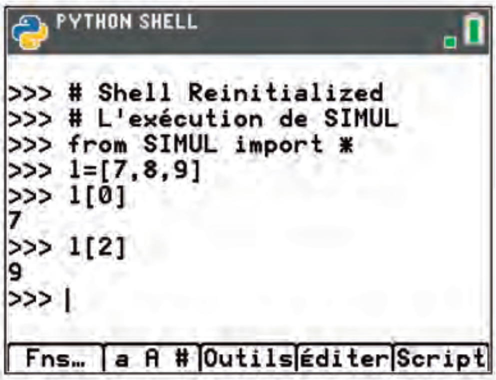


On exécute de nouveau notre script et on répète la démarche précédente.


Cette fois-ci, on obtient comme réponse `7.0` à l'instruction `sqrt(49)`.


On retourne dans notre script et on importe, de la même manière, la librairie `random` à partir, cette fois-ci, du module `2:random…`

## 2. Créer une liste




Une liste est un ensemble fini de valeurs que l'on souhaite stocker « au même endroit ». On utilise le crochet ouvrant `[` pour ouvrir notre liste et le crochet fermant `]` pour la refermer. Chaque élément est séparé par une virgule `,` Le premier élément de la liste porte l'indice 0, le second 1 etc.

Exécutons et définissons une liste `l` à l'aide de l'instruction : `l = [7,8,9]`

`[` ou `]` : touche [2nde] + [×] ou [−] et = : touche [sto→]

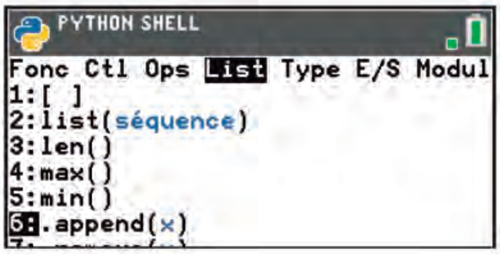
Puis saisissons l'instruction `l[0]`. La console retourne 7, la première valeur (indice 0) contenue dans la liste `l` que nous avons créée.

Saisissons l'instruction `l[2]`. La console retourne 9, la troisième valeur (indice 2) contenue dans la liste `l`.

# Importer des librairies
# Utiliser les listes



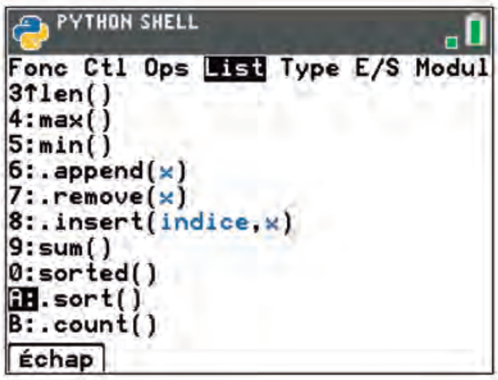
## 3. Utiliser les méthodes pour les listes




On commence par saisir `l` le nom de notre liste puis retournons dans Fns… et on se rend dans l'onglet `List`

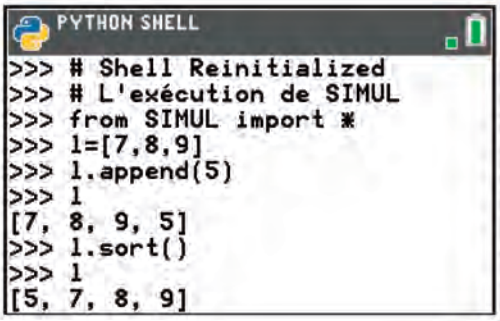
De nombreuses méthodes sont disponibles pour manipuler les listes

Par exemple sélectionnons l'item `6:.append(x)`



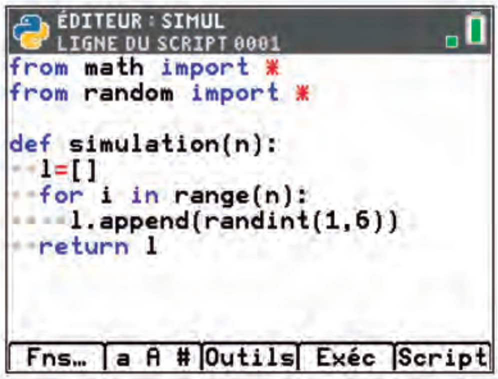

On complète l'instruction pour obtenir `l.append(5)`. On valide. L'élément 5 a été ajouté à la liste `l`.

On reprend la manipulation pour utiliser l'item `A:.sort()`. On valide. La liste `l` a été triée dans l'ordre croissant.




## 4. Réaliser une simulation de lancer de dés

On complète le script `SIMUL` avec la fonction `simulation` qui prend le nombre de lancers `n` en paramètre et renvoie la liste `l` contenant chacun des résultats de lancer du dé.




On utilise la fonction `randint` de la librairie `random`. Elle renvoie un nombre entier aléatoire compris entre les deux nombres entiers passés en paramètre.

Avec l'instruction `randint(1,6)`, on simule le lancer d'un dé à six faces.

A l'aide de la structure `for`, indentée correctement et l'utilisation notamment de l'instruction `range(n)`, on répète les lancers $n$ fois.

Chaque résultat renvoyé par la fonction `randint(1,6)` est ajouté dans la liste `l` à l'aide de l'instruction `l.append(…)`

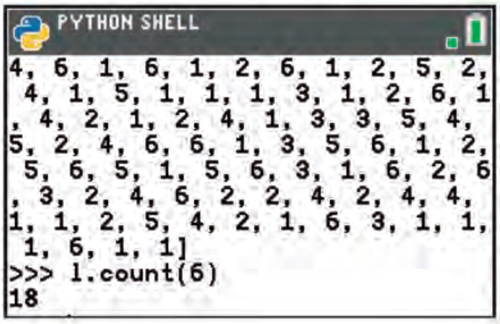
Il n'y a plus qu'à tester notre fonction dans la console.

Pour cela on exécute Exéc


On saisit l'instruction `l=simulation(100)` pour sauvegarder le résultat dans une liste `l`. La touche var permet de sélectionner le nom de la fonction sans avoir à le saisir. C'est un gain de temps. On valide.

On saisit `l` et on valide pour obtenir l'affichage du contenu de la liste `l`.

On compte le nombre de 6 obtenus à l'aide de l'instruction `l.count(6)` (item `B:.count()` dans l'onglet `List` via Fns…)




Dans notre simulation, nous avons obtenu 18 fois la face 6.

Thème: Programmation Python.

# Indentation – Si, alors, sinon

TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

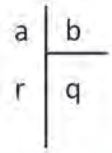
## Notre année est-elle bissextile ?

On va progressivement construire un script qui va nous permettre de savoir si une année est bissextile ou pas.

La rotation de la Terre autour du Soleil se fait en un peu plus de 365 jours. Ainsi, si notre calendrier contenait 365 jours, un décalage d'une journée apparaitrait (presque) tous les quatre ans.

<u>Règle n°1</u> : On considère que la rotation de la Terre autour du Soleil est de 365,25 jours. Ainsi tous les 4 ans on ajoute 1 jour dans le calendrier (c'est le 29 février). On a choisi d'ajouter ce jour si l'année est divisible par 4.

**Définition** : Soit $a$ et $b$ deux entiers non nuls. On effectue la division euclidienne de $a$ par $b$ :

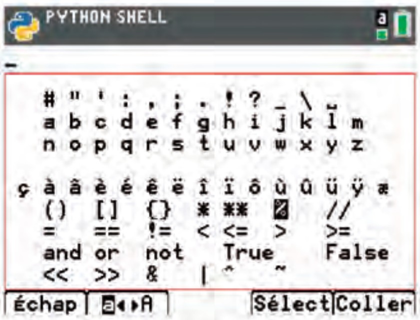



$q$ est appelé le quotient et $r$ le reste.

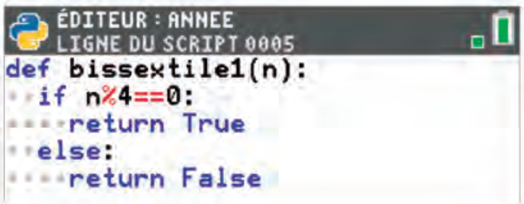
En Python on obtient $q$ en écrivant **a//b** et $r$ en écrivant **a%b**.

**Exemple** : En console (voir ci-contre) on obtient que le reste de la division euclidienne de 2022 par 4 est 2 et le reste de la division euclidienne de 2024 par 4 est 0. Ainsi 2022 n'est pas bissextile et 2024 est bissextile.


Pour obtenir le symbole % on appuie sur [a A #] puis on sélectionne le caractère % en appuyant sur **Sélect** puis on colle le caractère à l'aide de **Coller**.




Ecrire une fonction **bissextile1** qui prend comme argument un entier **n** représentant l'année et qui renvoie **True** si l'année **n** est bissextile et **False** sinon.




**True** et **False** sont accessibles dans [Fns…] **Ops** (tout en bas de la liste).

**Application** : Les années suivantes sont-elles bissextiles ?
2022,2024.

Après avoir exécuté notre script on appelle notre fonction **bissextile1** en appuyant sur [var] et on retrouve bien (voir ci-contre) que 2022 n'est pas bissextile mais 2024 est bissextile.

# Indentation – Si, alors, sinon



## Bien identer son script

Règle 2 : Une valeur plus précise du temps de la rotation de la Terre autour du Soleil est 365,24= $365 + \frac{1}{4} - \frac{1}{100}$

Ainsi on établit la règle suivante : l'année est bissextile si elle est divisible par 4 sauf si c'est une année de début de siècle (c'est-à-dire si elle divisible par 100).

Voici ci-contre le script de la fonction **bissextile2** qui prend en compte ces changements mais qui n'est pas indenté. A vous de l'indenter correctement !

```
def bissextile2(n) :
if n%4==0 :
if n%100 !=0 :
return True
else :
return False
else :
return False
```

Application : L'année 2100 est-elle bissextile ?

La bonne indentation est donnée dans l'exemple ci-contre.

```
ÉDITEUR : ANNEE
LIGNE DU SCRIPT 0016
def bissextile2(n):
  if n%4==0:
    if n%100!=0:
      return True
    else:
      return False
  else:
    return False
Fns… | a A # | Outils | Exéc | Script
```

A noter que « différent de 0 » s'écrit en Python `!=0` accessible dans `Fns…` `Ops`. On trouve que 2100 n'est pas bissextile.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de ANNEE
>>> from ANNEE import *
>>> bissextile2(2100)
False
```

En réalité une année dure 365,2425 jours.

Or $365{,}2425 = 365 + \frac{1}{4} - \frac{1}{100} + \frac{1}{400}$

La règle (définitive) retenue est la suivante :

Règle 3 : Une année est bissextile si elle est divisible par 4 sauf si c'est une année de début de siècle à moins qu'elle ne soit divisible par 400.

Autrement dit pour les années de début de siècle il faut qu'elle soit divisible par 400 pour être bissextile.

Voici ci-contre le script de la fonction **bissextile** qui prend en compte ces changements mais qui n'est pas indenté. A vous de l'indenter correctement !

```
def bissextile(n) :
if n%4==0 :
if n%100 !=0 :
if n%400 ==0 :
return True
else :
return False
else :
return True
else :
return False
```

Application : Les années 2000 2100 sont-elles bissextiles ?

On trouve que 1900 n'est pas bissextile mais 2000 est une année bissextile.

```
ÉDITEUR : ANNEE
LIGNE DU SCRIPT 0026
def bissextile(n):
  if n%4==0:
    if n%100==0:
      if n%400==0:
        return True
      else:
        return False
    else:
      return True
  else:
    return False
Fns… | a A # | Outils | Exéc | Script
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de ANNEE
>>> from ANNEE import *
>>> bissextile(1900)
False
>>> bissextile(2000)
True
```

# Prendre l'opposé ou soustraire ?

Il ne faut pas confondre le symbole [(-)] qui sert à définir l'opposé d'un nombre avec le symbole [−] qui sert à faire une soustraction.

## Dans un calcul

Exemple 1 : Je souhaite calculer $8 - 5$ :

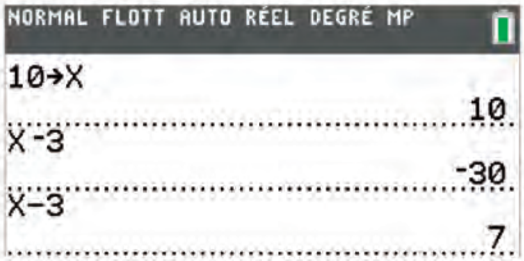
Le symbole - correspond à celui de la soustraction, il s'agit de la touche [−]

Tandis que le symbole [(-)] correspond à l'opposé, ici l'opposé de 5.


Pour comprendre ce qui se passe, il faut se rappeler que la multiplication en mathématiques peut être sous entendue. Utilisons une variable pour mieux comprendre :

$7X$ signifie $7 \times X$ de même, $X5$ signifie $X \times 5$ (même si on préfère écrire $5X$).

D'autre part, $-5X = X \times (-5)$. En écrivant **X-5** la calculatrice comprend que vous souhaitez multiplier **X** par **-5**.




De la même façon 8 [(-)] 5 va signifier $8 \times (-5) = -40$

Et 8 [−] 5 correspond à la soustraction : $8 - 5 = 3$

Exemple 2 : Soit $X = 10$, je souhaite calculer $3 - X$ :

Si j'écris $X$ [(-)] 3 alors cela correspond à $X \times (-3)$ soit $-3X$ qui vaut $-30$.

Dans notre cas, il faut écrire $X$ [−] 3 pour obtenir 7.

De même si on écrit **2X** ou **X2** la machine fera la même chose : elle va multiplier $X$ par 2.

Exemple 3 : Je souhaite calculer $-3 + 10$.

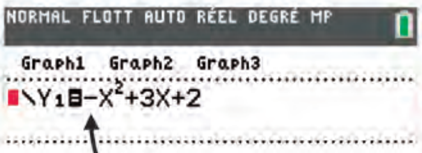
Pour additionner l'opposé de 3 avec 10 on va donc écrire [(-)] 3 +10.


## Représentation graphique

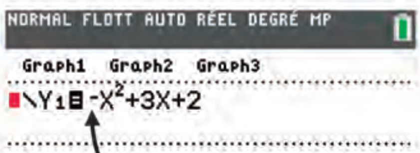
Si on souhaite entrer l'expression d'une fonction $f$ définie sur $\mathbb{R}$ par

$f(x) = -x^2 + 3x + 2$ alors le signe - correspond à l'opposé de $X$ : [(-)]

Si on utilise le signe [−] cela va créer le message d'erreur ci-contre :


Erreur




Correct

# Tips and tricks

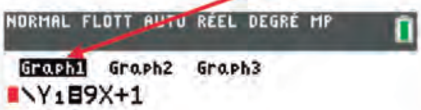
TI-82 Advanced
Edition Python

TI-83 Premium CE
Edition Python

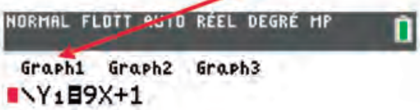
## Erreurs et graphiques

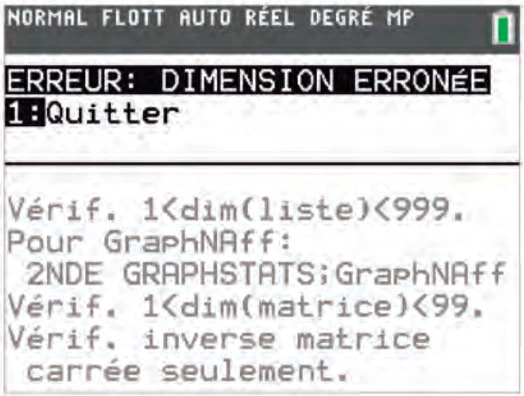
Lorsqu'on souhaite représenter graphiquement une fonction, il faut désactiver les autres graphiques issus des statistiques.

**Graph1** est activé.




**Graph1** est désactivé.



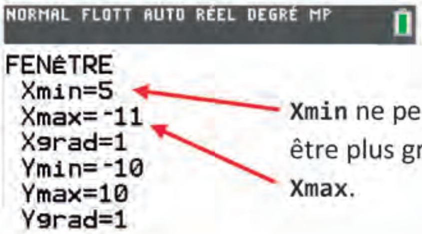

En représentant graphiquement la fonction dont l'expression est stockée dans **Y1**, étant donné que **Graph1** est activé vous allez obtenir deux représentations graphiques (une courbe et un nuage de point par exemple).

Mais si entre temps les listes auxquels **Graph1** fait référence sont effacées alors vous rencontrerez un message d'erreur (ci-contre).

Il faut désactiver **Graph1** en se déplaçant tout simplement dessus avec le curseur et valider en appuyant sur [entrer].

## Fenêtre incohérente

Dans la fenêtre de représentation graphique, si **Xmin** et **Xmax** ne sont pas cohérents cela provoquera une erreur.



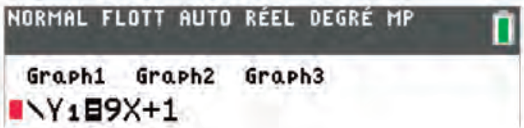

**Xmin** ne peut pas être plus grand que **Xmax**.


## Y1 ou Y1 ?

Dans une page de calcul, lorsqu'on souhaite calculer **Y1(5)** il ne faut pas taper **Y** puis **1**, mais utiliser [var] **VAR Y** puis **Fonctions**.

Y1(5) renvoie le calcul : Y× 1 × 5, et ne calcule pas l'image de 5.


**Y1(5)** calcule bien l'image de 5 par la fonction **Y1**.



## Plus d'informations sur le site internet de T³ France :

- Des ressources pédagogiques pour votre classe
- Un programme de formations gratuites sur site et en ligne
- Des vidéos d'aide à la prise en main de la technologie

**t3france.fr**

Un service après-vente est également accessible depuis le site **education.ti.com/fr/csc**