

Utilisation de la Casio Graph90+E



- ▶ **Étude de suites** Fiches 1 et 2
- ▶ **Étude de fonctions** Fiches 3 et 4
- ▶ **Tableur** Fiche 5
- ▶ **Probabilités et statistiques** Fiche 6
- ▶ **Programmation Python** Fiche 7

Ces fiches sont issues du site www.casio-education.fr.

Vous pouvez également consulter la [chaîne YouTube Casio Education](#).

Suites

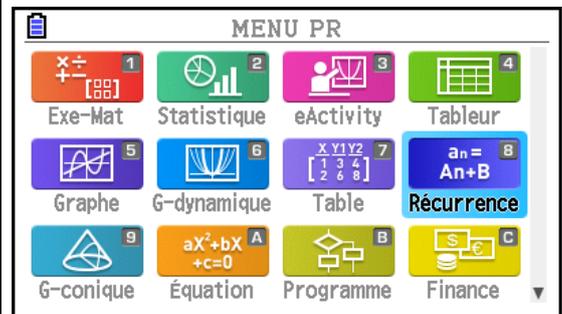


1.	Accès au menu.....	1
2.	Formule explicite	1
3.	Relation de récurrence	5

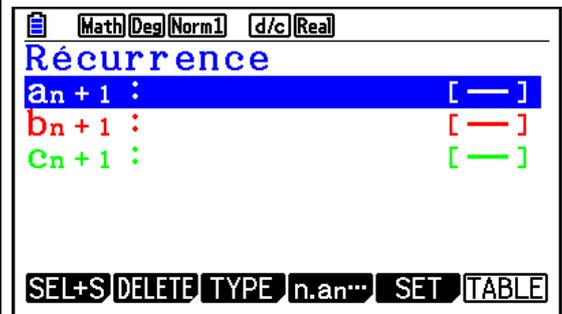
1. Accès au menu

À partir de la liste des menus, sélectionner à l'aide du pavé directionnel, le menu **Récurrence**. Valider en saisissant la touche **EXE**.

 Appuyer sur la touche **8** pour accéder directement au menu **Récurrence** (Graph 90+E), **RECUR** (Graph 35+E). Ce menu n'est pas disponible avec la calculatrice Graph 25+E).



Le menu **Récurrence** s'affiche.

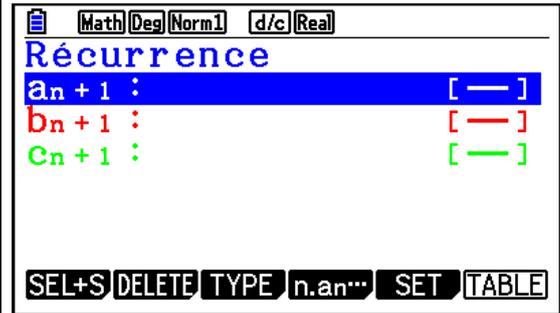


2. Formule explicite

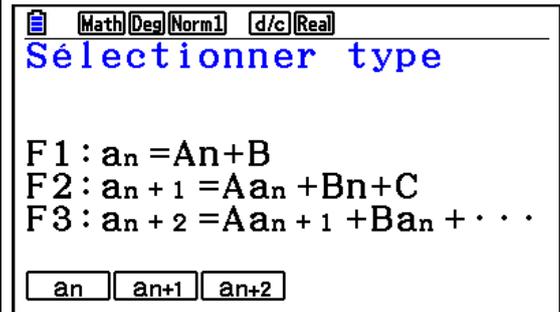
Application : soit la suite (a_n) définie, pour $n \in \mathbb{N}$, par : $a_n = 2 \times 1,5^n - 1$

- 1) Déterminer les 16 premiers termes.
- 2) Déterminer la somme des 16 premiers termes.
- 3) Représenter graphiquement le nuage de points des 16 premiers termes de la suite (a_n) .

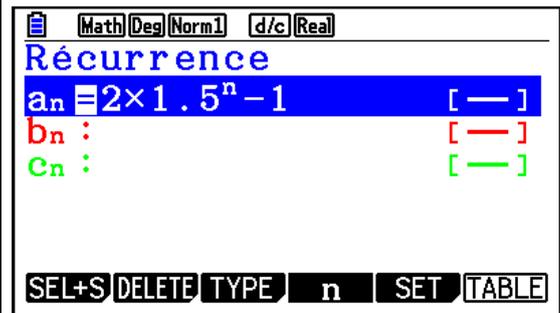
1) À partir du Menu **Récurrance**, appuyer sur la touche **F3** {TYPE}.



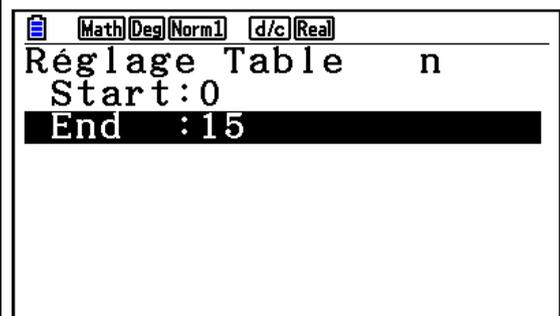
- Saisir une suite définie par une formule explicite :
Presser la touche **F1** { a_n }.



Saisir ensuite la formule suivante : $2 \times 1,5^n - 1$
Pour obtenir la variable n, appuyer sur **F4** { $n.a_n...$ }
puis **F1** {n}.
Valider en pressant la touche **EXE**.

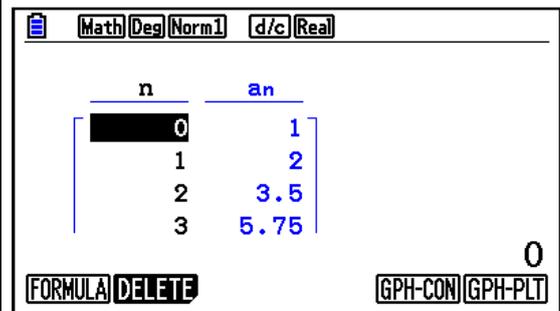


- Saisir la plage du tableau de valeurs :
Appuyer sur la touche **F5** {SET} afin d'indiquer la valeur initiale ("0") et la valeur finale ("15") pour n.
Valider avec la touche **EXE**.



- Afficher le tableau de valeurs :
Vérifier que seule la ligne où se trouve l'expression de la suite dont on souhaite éditer les termes possède un signe = en surbrillance.

Appuyer sur la touche **F6** {TABL}.
Nous obtenons ainsi les 16 premiers termes de la suite.





A l'aide du menu **Exe-Mat** (Graph 90+E), **RUN MAT** (Graph 35+E), il est aussi possible de déterminer les 16 premiers termes de la suite (a_n).

Utiliser la syntaxe suivante :

Seq(expression, variable, début, fin, pas)

$Seq(2 \times 1.5^n - 1, N, 0, 1, 15, 1)$

Appuyer sur les touches **[OPTN]**, **[F1]** {List} et **[F5]** {Seq} pour accéder à Seq.

2) Effectuer la somme des p premiers termes :

- 1^{ère} méthode :

Appuyer sur les touches **[SHIFT]** **[MENU]** pour accéder à **[SET UP]**.

A l'aide du pavé directionnel, se déplacer jusqu'à la ligne " Σ Display".

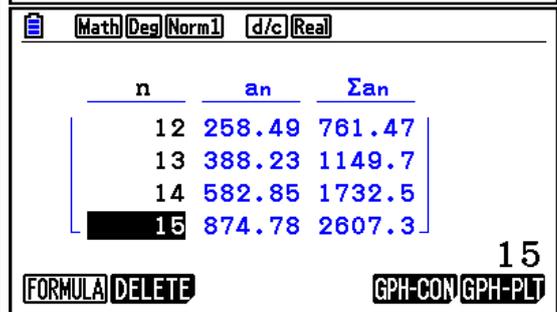
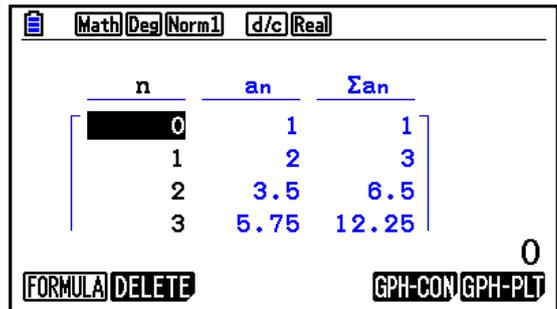
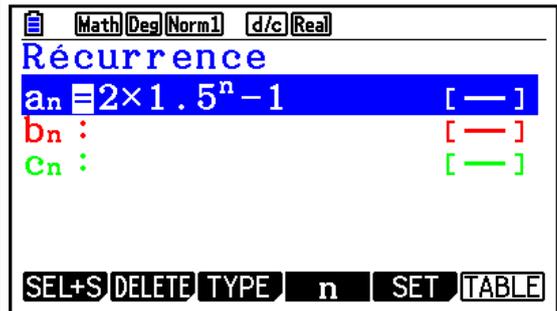
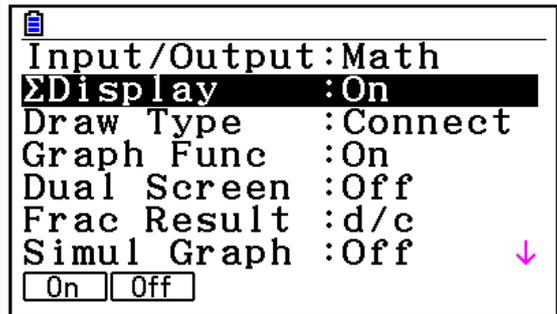
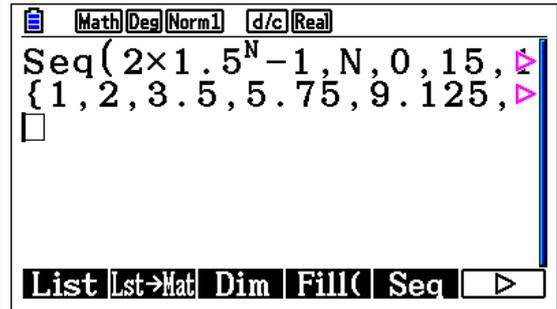
L'affichage par défaut est en mode OFF.

Afin d'afficher la somme des termes de la suite, saisir la touche **[F1]** {On}.

Appuyer ensuite sur la touche **[EXIT]** pour revenir au menu **Réccurrence** (Graph 90+E), **RECUR** (Graph 35+E).

Presser la touche **[F6]** {TABL} pour accéder au tableau de valeurs.

La somme des 16 premiers termes est d'environ 2 607,3.



- 2^{ème} méthode :

À partir du Menu **Exe-Mat** (Graph 90+E), **RUN-MAT** (Graph 35+E), appuyer sur la touche **OPTN**.

Saisir ensuite les touches **F4** {CALC}, **F6** {} et **F3** {Σ}.

Ecrire la formule suivante : $\sum_{N=0}^{15} 2 \times 1,5^N - 1$

Valider avec la touche **EXE**

La somme des 16 premiers termes est d'environ 2 607,3.

3) Représentation graphique - Nuage de points :
 Dans le menu **Récurrance** (Graph 90+E), **RECUR** (Graph 35+E), appuyer sur les touches **SHIFT** **F3** [**V-WINDOW**] pour accéder aux paramètres d'affichage graphique.

 [**V-WINDOW**] permet de définir les valeurs minimales et maximales de x et de y ainsi que l'échelle de graduations de chaque axe X scale et Y scale.

Nous allons choisir les paramètres suivants :

$$-1 \leq x \leq 16 \quad -10 \leq y \leq 900$$

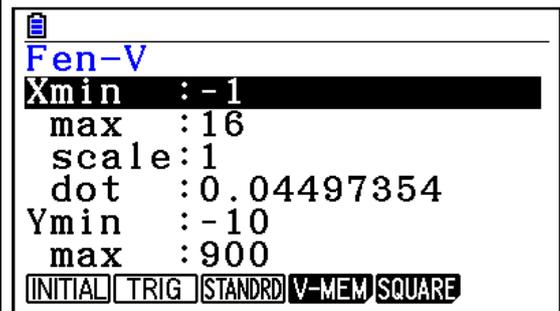
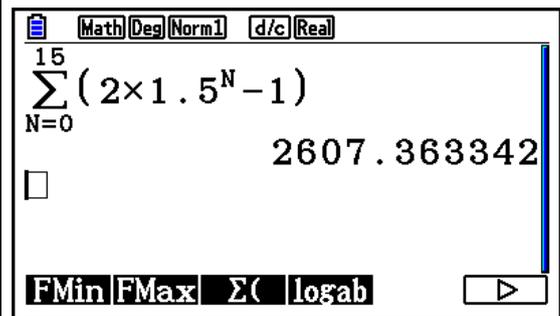
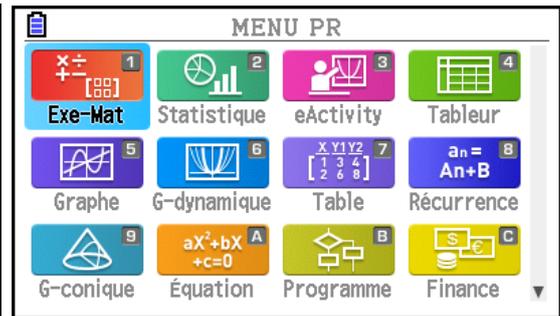
Graduation de 1 sur l'axe des abscisses (X scale).

Graduation de 50 sur l'axe des ordonnées (Y scale).

La variable *Xdot* est automatiquement recalculée par la calculatrice en fonction des valeurs entrées pour *Xmin* et *Xmax*.

Après chaque saisie, presser la touche **EXE**.

Appuyer sur la touche **EXIT** pour revenir à l'éditeur de suites.



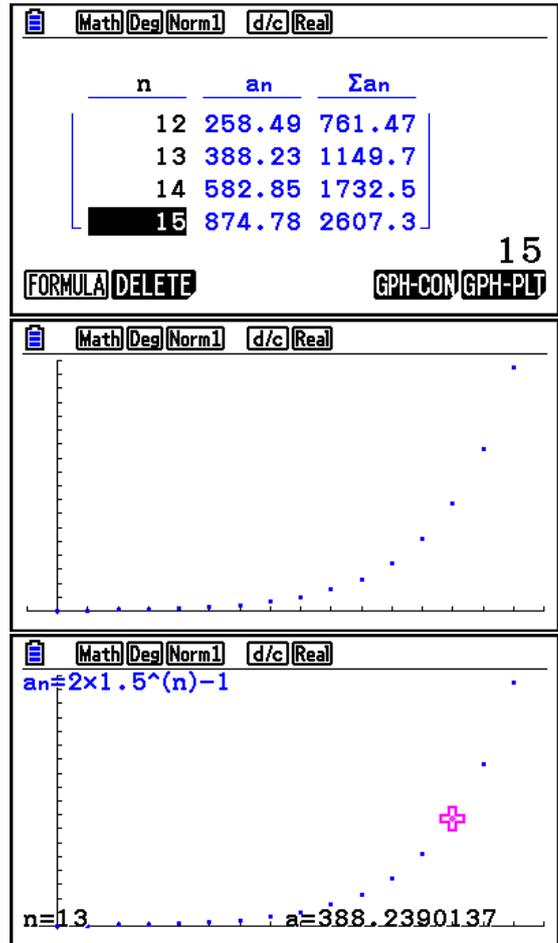
Presser la touche **[F6]** {TABL} pour accéder au tableau, **[F6]** pour éditer le nuage de points des 16 premières valeurs de la suite (a_n) {GPH-PLT}.

La touche **[F1]** [TRACE] permet d'éditer les coordonnées des 16 premières valeurs de la suite (a_n) .



Le curseur se déplace automatiquement sur le premier point solution, le plus à gauche de l'écran.

Déplacer la croix sur la courbe à l'aide des flèches **[◀]** **[▶]** pour obtenir les autres points.

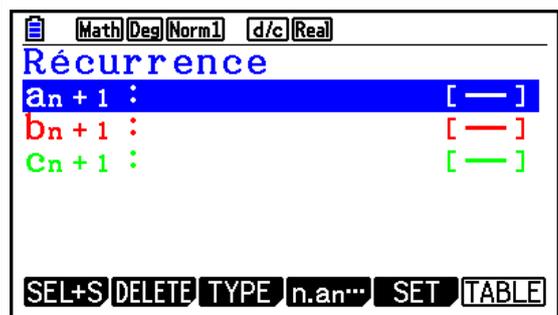


3. Relation de récurrence

Application : soit la suite (a_n) définie par $\begin{cases} a_{n+1} = 2n - a_n \\ a_0 = -2 \end{cases}$ pour $n \in \mathbb{N}$.

- 1) Déterminer les 16 premiers termes.
- 2) Déterminer la somme des 16 premiers termes.
- 3) Représenter graphiquement le nuage de points des 16 premiers termes de la suite (a_n) .

1) À partir du Menu **Récurrence**, appuyer sur la touche **[F3]** {TYPE}.



- Saisir une suite définie par récurrence :
Presser la touche **F2** { a_{n+1} }.

Puis, saisir la formule suivante : $a_{n+1} = 2n - a_n$
Pour obtenir la variable n, appuyer sur **F4** {n.a_n...}
puis **F1** {n} ou **F2** {a_n}.
Valider en pressant la touche **EXE**.

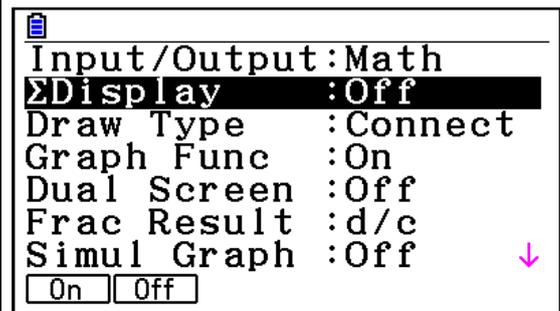
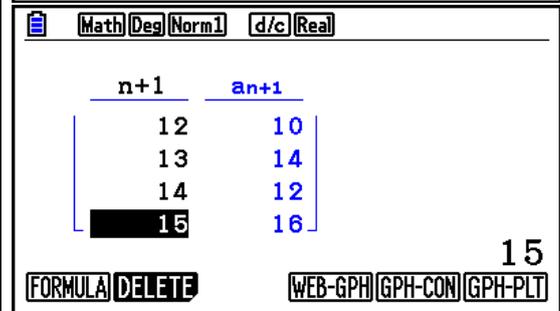
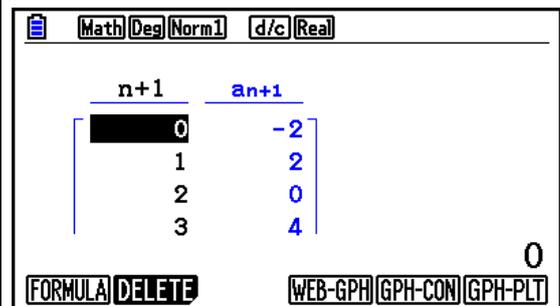
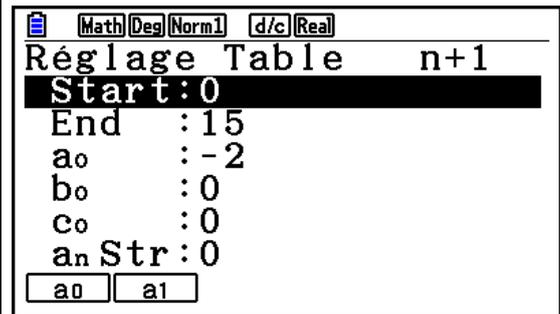
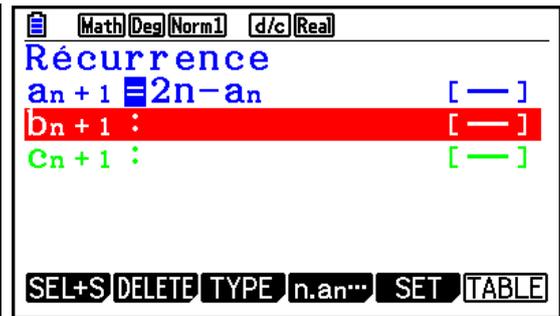
- Saisir la plage du tableau de valeurs :
Appuyer sur la touche **F5** {SET} pour indiquer le premier terme ("a₀ = -2"), la valeur initiale ("0") et la valeur finale ("15") pour n.
Valider avec la touche **EXE**.

- Afficher le tableau de valeurs :
Vérifier que seule la ligne où se trouve l'expression de la suite dont on souhaite éditer les termes possède un signe = en surbrillance.

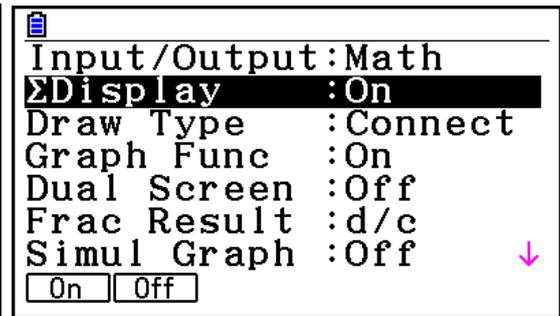
Appuyer sur la touche **F6** {TABL}.
Nous obtenons ainsi les 16 premiers termes de la suite.

2) Effectuer la somme des p premiers termes :

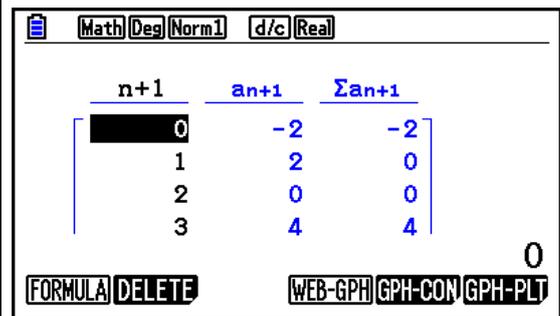
- 1^{ère} méthode :
Appuyer sur les touches **SHIFT** **MENU** pour accéder à **[SET UP]**.
A l'aide du pavé directionnel, se déplacer jusqu'à la ligne "Σ Display".
L'affichage par défaut est en mode OFF.
Afin d'afficher la somme des termes de la suite, saisir la touche **F1** {On}.
Appuyer ensuite sur la touche **EXIT** pour revenir au menu **Récurrence**



Presser la touche **F6** {TABL} pour accéder au tableau de valeurs.

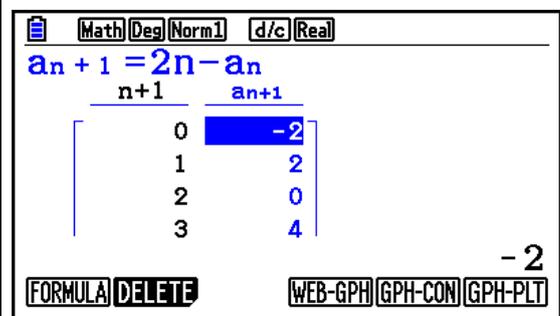
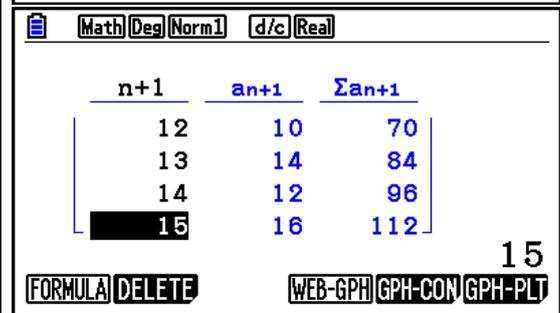


La somme des 16 premiers termes est de 112.

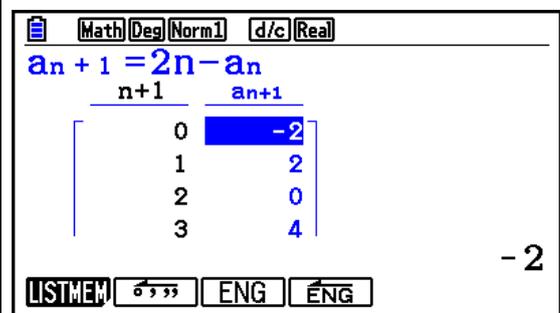


- 2^{ème} méthode :

A partir du tableau de valeurs, se positionner à l'aide du pavé directionnel sur une valeur de la colonne a_{n+1} .



Appuyer sur les touches **OPTN** puis **F1** {LISTMEM}.



Presser ensuite la touche **[1]** pour enregistrer les données de la colonne a_{n+1} dans la liste 1. Valider en saisissant la touche **[EXE]**.

Les données sont aussi stockées dans l'éditeur de listes (colonne List 1) du menu **Statistique** (Graph 90+E), **STAT** (Graph 35+E).

Se déplacer ensuite dans le menu **Exe-Mat** (Graph 90+E), **RUN MAT** (Graph 35+E).

Puis, saisir les touches **[OPTN]**, **[F1]** {LIST}. Entrer ensuite la formule suivante : Sum List 1. Pour accéder aux commandes List et Sum, presser la touche **[F1]** {LIST} et **[F6]** {**[▶]**} deux fois puis **[F1]** {SUM}.

La somme des 16 premiers termes est de 112.

3) Représentations graphiques :

Dans le menu **Récurrance**, appuyer sur les touches **[SHIFT]** **[F3]** **[V-WINDOW]** pour accéder aux paramètres d'affichage graphique.

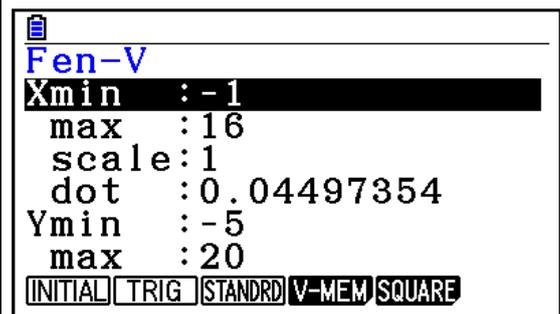
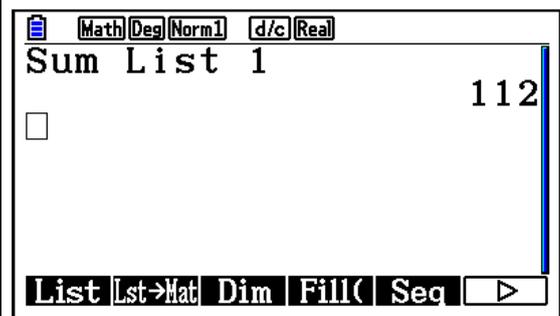
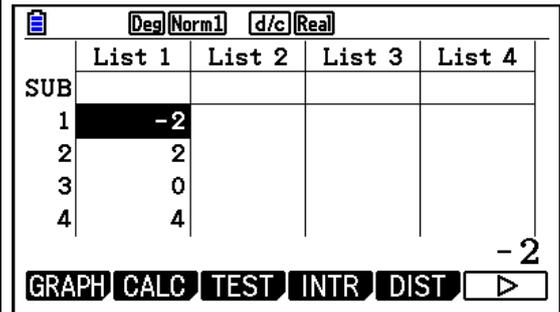
Nous allons choisir les paramètres suivants :

$$-1 \leq x \leq 16 \quad -5 \leq y \leq 20$$

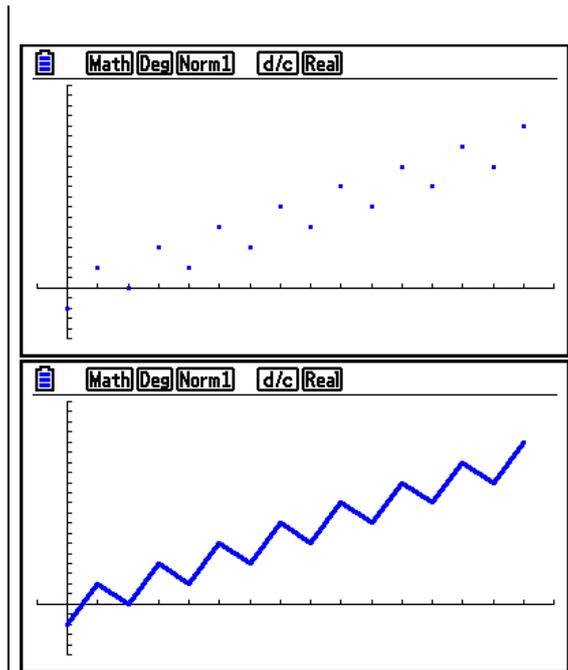
Graduation de 1 sur l'axe des abscisses (*X scale*).

Graduation de 1 sur l'axe des ordonnées (*Y scale*).

Après chaque saisie, presser la touche **[EXE]**.



Presser la touche **F6** {TABL} pour accéder au tableau, puis **F6** {GPH-PLT} pour éditer le nuage de points, ou **F5** {GPH-CON} pour éditer la représentation graphique des 16 premières valeurs de la suite (a_n).

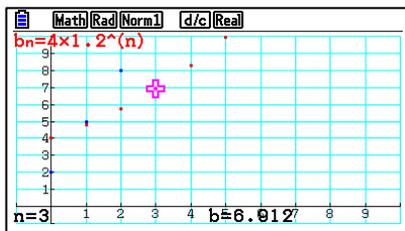


Fiche pratique : SUITES, Menu Récurrence

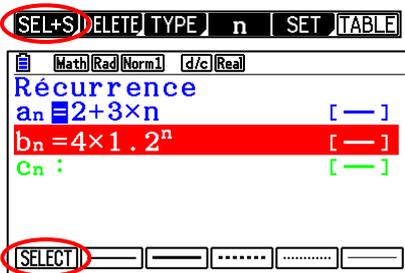


Menu Récurrence

<p>Math Rad Norm1 d/c Real Récurrence a_{n+1} : [-] b_{n+1} : [-] c_{n+1} : [-] SEL+S DELET TYPE n.an... SET TABLE</p>	<p>Pour modifier le type de formules utilisées, on utilise la touche F3 {TYPE}.</p>															
<p>Math Rad Norm1 d/c Real Sélectionner type F1 : a_n = A_n + B F2 : a_{n+1} = A a_n + B_n + C F3 : a_{n+2} = A a_{n+1} + B a_n + ... a_n a_{n+1} a_{n+2}</p>	<p>On peut alors utiliser la touche F1 {a_n}, pour entrer des formules de suites sous forme explicite.</p>															
<p>Math Rad Norm1 d/c Real Récurrence a_n = 2 + 3 × n [-] b_n = 4 × 1.2ⁿ [-] c_n : [-] SEL+S DELET TYPE n SET TABLE</p>	<p>On peut entrer les formules des suites à étudier, ici :</p> <ul style="list-style-type: none"> - une suite (a_n) arithmétique : a_n = 2 + 3 × n - une suite (b_n) géométrique : b_n = 4 × 1,2ⁿ <p>En utilisant la touche F5 {SET}, on peut alors régler les valeurs minimale et maximale de n.</p>															
<p>Math Rad Norm1 d/c Real Réglage Table n Start : 0 End : 30</p>	<p>Ici les valeurs choisies pour n sont 0 pour la valeur minimale et 30 pour la valeur maximale.</p> <p>En utilisant la touche EXIT, on revient alors à la fenêtre précédente.</p>															
<p>SEL+S DELET TYPE n SET TABLE</p> <p>Math Rad Norm1 d/c Real</p> <table border="1"> <thead> <tr> <th>n</th> <th>a_n</th> <th>b_n</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>4</td> </tr> <tr> <td>1</td> <td>5</td> <td>4.8</td> </tr> <tr> <td>2</td> <td>8</td> <td>5.76</td> </tr> <tr> <td>3</td> <td>11</td> <td>6.912</td> </tr> </tbody> </table> <p>FORMULA DELET PHASE GPH-CO GPH-PLT</p>	n	a _n	b _n	0	2	4	1	5	4.8	2	8	5.76	3	11	6.912	<p>La touche F6 {TABLE} donne accès aux valeurs prises pour chaque terme des suites (a_n) et (b_n).</p> <p>Remarque pour la Graph 35+E : F6 {TABL}.</p> <p>En utilisant le pavé directionnel, on peut lire toutes ces valeurs.</p> <p>La touche F6 {GPH-PLT} permet de faire apparaître le graphique représentant les termes de suites (a_n) et (b_n) par des points de couleurs.</p> <p>Remarque pour la Graph 35+E : F6 {G-PLT}.</p>
n	a _n	b _n														
0	2	4														
1	5	4.8														
2	8	5.76														
3	11	6.912														

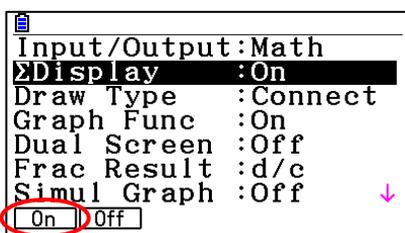


On peut alors sélectionner **F1** (**Trace**), ce qui permet, à l'aide du pavé directionnel, de retrouver en bas de l'écran le rang du terme, sa valeur et en haut le nom de la suite et sa formule. En appuyant deux fois sur la touche **EXIT**, on revient à la fenêtre de définition des suites.



La touche **F1** {SEL+S} permet de sélectionner ou non la ou les suites que l'on souhaite étudier en validant à nouveau sur **F1** {SELECT}.

En utilisant la touche **EXIT**, on revient alors à la fenêtre précédente.



En utilisant les touches **SHIFT** puis **MENU** (**SET UP**), on peut ensuite effectuer les sommes des termes des suites.

Il faut pour cela, à l'aide de la touche **F1** {On}, valider le Σ Display.

En utilisant la touche **EXIT**, on revient alors à la fenêtre précédente.

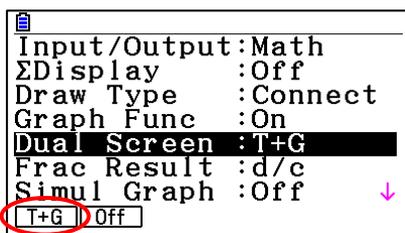
n	a_n	Σa_n
0	2	2
1	5	7
2	8	15
3	11	26

On utilise ensuite la touche **F6** {TABLE}.

Une nouvelle colonne Σa_n apparaît alors dans la table.

On y retrouve les sommes successives des premiers termes de la suite (a_n).

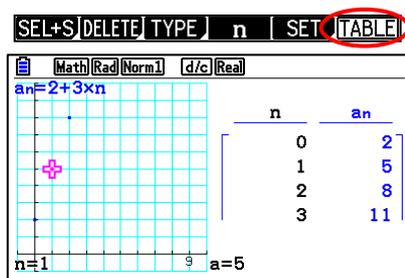
Remarque pour la Graph 35+E : **F6** {TABL}.



La visualisation simultanée de la table et du graphique est possible.

Pour cela, dans le **SET UP** (touches **SHIFT** puis **MENU**), il faut régler le Dual Screen à l'aide de la touche **F1** {T+G} et mettre le Σ Display sur {Off}.

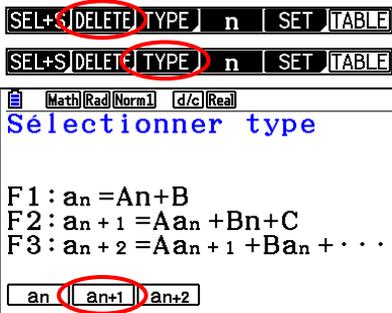
En utilisant la touche **EXIT**, on revient alors à la fenêtre de définition des suites.



On peut alors utiliser la touche **F6** {TABLE} pour faire apparaître la table et le graphique.

Sélectionner **F1** {Trace}, ce qui permet, à l'aide du pavé directionnel, de retrouver en bas de l'écran le rang du terme, sa valeur et en haut le nom de la suite et sa formule.

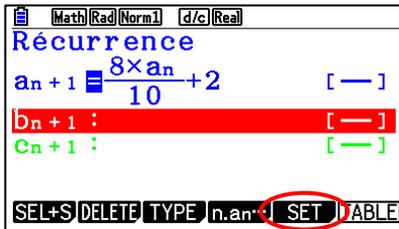
Remarque pour la Graph 35+E : **F6** {TABL}.



On veut maintenant étudier une suite donnée par une formule de récurrence.

Après avoir effacé les anciennes formules dans la fenêtre d'accueil à l'aide de la touche **[F2]** {DELETE}, il faut modifier le type de formules utilisées avec la touche **[F3]** {TYPE}.

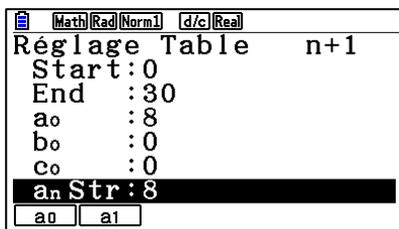
On sélectionne **[F2]** {a_{n+1}}, pour entrer des formules de suites avec une formule de récurrence.



On peut entrer la formule de récurrence de la suite à étudier :
- une suite (a_n) arithmético-géométrique : a_{n+1}=0,8×a_n+2 ; a₀=8

Pour la suite, il sera intéressant d'entrer la formule sous forme fractionnaire.

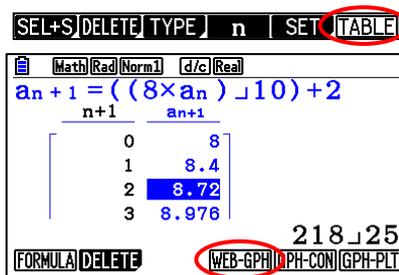
En utilisant la touche **[F5]** {SET}, on peut alors régler des valeurs.



Ici, les valeurs choisies pour n sont 0 pour la valeur minimale, 30 pour la valeur maximale et 8 pour la valeur du premier terme de la suite a₀.

On réglerait aussi a_nStr sur 8 pour voir la construction graphique de chaque terme de la suite (a_n).

En utilisant la touche **[EXIT]**, on revient alors à la fenêtre précédente.



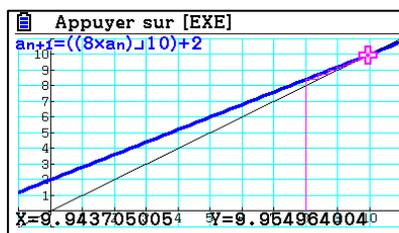
On utilise ensuite la touche **[F6]** {TABLE} pour obtenir la table de valeurs de la suite (a_n).

Remarque pour la Graph 35+E : **[F6]** {TABL}.

A l'aide du pavé directionnel, il est possible de se déplacer sur les termes de la suite ; l'intérêt d'avoir entré la formule de récurrence sous forme fractionnaire permet de retrouver la valeur exacte du terme (en bas à droite de l'écran).

Pour visualiser la construction des termes de la suite (a_n), il faut utiliser la touche **[F4]** {WEB-GPH}.

Remarque pour la Graph 35+E : **[F4]** {WEB}.

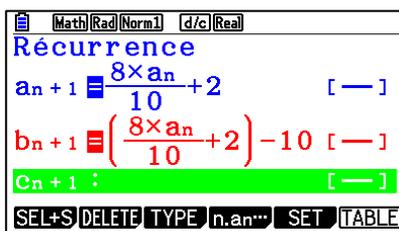


Sur le graphique on obtient :

- la courbe représentative de la fonction $f(x) = 0,8x + 2$
- la courbe représentative de la fonction $f(x) = x$

Par pression successive de la touche **[EXE]**, on voit se construire les termes de la suite (a_n) en partant de la valeur a₀.

On observe aussi que la suite semble converger vers 10, valeur dont on va pouvoir se servir ensuite.



En pressant 2 fois la touche **[EXIT]**, on revient à la fenêtre des formules de récurrence.

On définit maintenant une nouvelle suite (b_n) :

$$b_{n+1} = 0,8 \times a_n + 2 - 10 ; b_0 = - 2$$

[SEL+S][DELETE] TYPE] n [SET] TABLE

Math Rad Norm1 d/c Real

Réglage Table n+1

Start: 0

End : 30

a₀ : 8

b₀ : -2

c₀ : 0

a_n Str : 0

a₀ a₁

En utilisant la touche [F5] {SET}, on peut régler les valeurs.

Ici les valeurs choisies pour n sont 0 pour la valeur minimale, 30 pour la valeur maximale, 8 pour la valeur du premier terme de la suite a₀ et -2 pour la valeur du premier terme de la suite b₀.

En utilisant la touche [EXIT], on revient à la fenêtre de définition des suites.

[SEL+S][DELETE] TYPE] n [SET] TABLE

Math Rad Norm1 d/c Real

$b_{n+1} = ((8 \times a_n) \downarrow 10) + 2 -$

n+1	a _{n+1}	b _{n+1}
0	8	-2
1	8.4	-1.6
2	8.72	-1.28
3	8.976	-1.024

-32.25

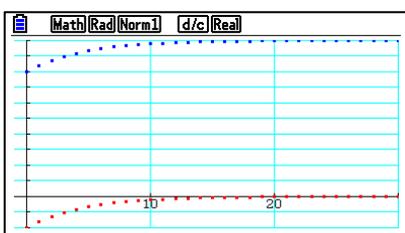
FORMULA DELETE PHASE WEB-GPH GPH-CON GPH-PLT

En pressant la touche [F6] {TABLE}, on peut afficher la table des valeurs pour (a_n) et (b_n).

Remarque pour la Graph 35+E : [F6] {TABL}.

Avec la touche [F6] {GPH-PLT}, on visualise ensuite les termes des deux suites dans la fenêtre graphique (on pourra remarquer que la suite (b_n) est une suite géométrique de raison 4/5).

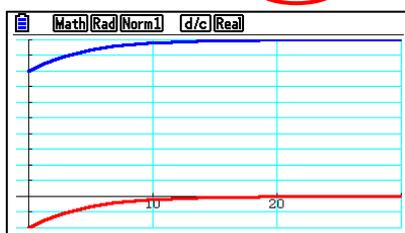
Remarque pour la Graph 35+E : [F6] {G-PLT}.



On observe bien le décalage en ordonnée de -10.

Si on souhaite mieux visualiser ces deux suites, il est possible d'utiliser les fonctions : on revient ainsi à la fenêtre précédente avec la touche [EXIT].

FORMULA DELETE PHASE WEB-GPH GPH-CON GPH-PLT



On utilise alors la touche [F5] {GPH-CON}.

Sur le graphique on obtient :

- la courbe représentative de la fonction $f(U_{n+1}) = 0,8U_n + 2$
- la courbe représentative de la fonction $f(V_n) = U_n - 10$

Remarque pour la Graph 35+E : [F5] {G-CON}.

Math Rad Norm1 d/c Real

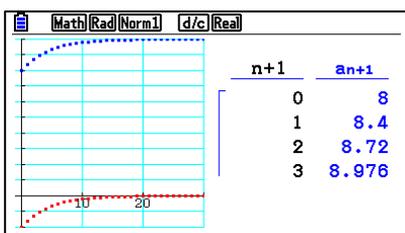
$a_{n+1} = ((8 \times a_n) \downarrow 10) + 2$

n+1	a _{n+1}	Σa _{n+1}	b _{n+1}
0	8	8	-2
1	8.4	16.4	-1.6
2	8.72	25.12	-1.28
3	8.976	34.096	-1.024

82.5

FORMULA DELETE PHASE WEB-GPH GPH-CON GPH-PLT

De la même manière, il sera possible de visualiser la somme des termes des suites (a_n) et (b_n) en entrant dans le **Set Up** de la calculatrice ([SHIFT] puis [MENU]) et en validant le ΣDisplay à l'aide de la touche [F1] {On}.

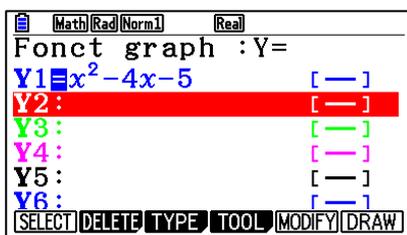


Il sera aussi possible de visualiser simultanément des tables et les représentations graphiques des termes des suites (a_n) et (b_n) en entrant dans le **Set Up** de la calculatrice ([SHIFT] puis [MENU]) et en réglant le Dual Screen à l'aide de la touche [F1] {T+G}.

Fiche pratique: TRACÉ DE COURBES



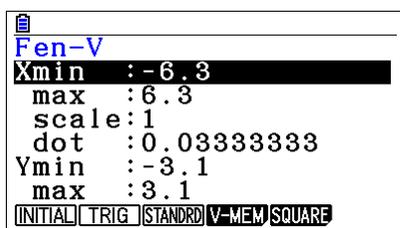
Menu Graph



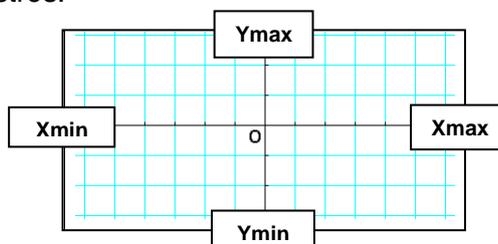
Utiliser la touche $\boxed{X, \theta, T}$ pour saisir x .

- $\boxed{F1}$ {SELECT} → Sélectionner/Désélectionner une ligne (expression sélectionnée: signe égal surligné)
- $\boxed{F2}$ {DELETE} → Supprimer une expression (on peut aussi utiliser \boxed{DEL})
- $\boxed{F3}$ {TYPE} → Changer le type d'expression (Y=;r=;...Y>;Y<;...X>;X<;...)
- $\boxed{F6}$ {DRAW} → Tracer la représentation graphique (on peut aussi utiliser \boxed{EXE})

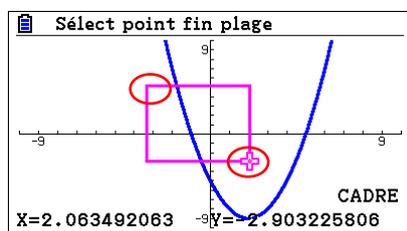
\boxed{SHIFT} $\boxed{F3}$ (V-Window)



$\boxed{F1}$ {INITIAL} / $\boxed{F2}$ {TRIG} / $\boxed{F3}$ {STANDARD} → 3 repères préenregistrés.

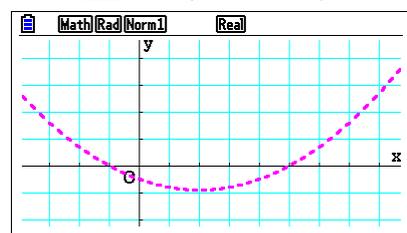


\boxed{SHIFT} $\boxed{F2}$ (Zoom)

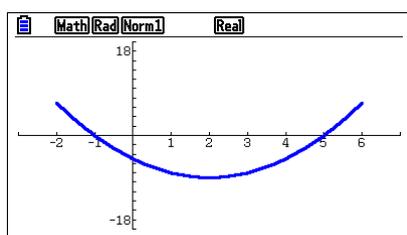
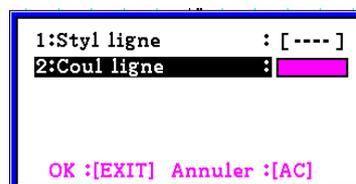


$\boxed{F1}$ {BOX} → Pour définir le cadre de zoom se déplacer sur l'un de ses sommets avec les flèches ou en sélectionnant directement l'abscisse du point au clavier de la calculatrice et appuyer sur \boxed{EXE} puis se déplacer sur le point diagonalement opposé et appuyer sur \boxed{EXE}

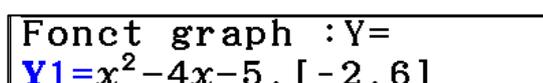
\boxed{SHIFT} $\boxed{5}$ (FORMAT)



Pour changer style et les couleurs:



Pour tracer la courbe sur un intervalle donné.



Fiche pratique: ÉTUDE DE FONCTION



Menu Graphe

- Pour changer de courbe utiliser les flèches \blacktriangle \blacktriangledown
- Pour aller sur un point particulier, se déplacer avec les flèches \blacktriangleleft \blacktriangleright , appuyer sur $\boxed{X\theta T}$ ou sélectionner directement l'abscisse du point avec le clavier de la calculatrice.

<p>SHIFT F1 (Trace)</p>	<p>Se déplacer sur le point souhaité avec les flèches ou sélectionner directement l'abscisse du point avec le clavier de la calculatrice.</p> <p>$\boxed{\text{EXE}}$ → Permet de fixer les coordonnées du point à l'écran (Disponible uniquement sur la Graph 90+E)</p>
<p>SHIFT F5 (G-Solv)</p> <p>$\boxed{\text{ROOT}}$ $\boxed{\text{MAX}}$ $\boxed{\text{MIN}}$ $\boxed{\text{Y-ICEPT}}$ $\boxed{\text{INTSECT}}$ \blacktriangleright</p> <p>$\boxed{\text{Y-CAL}}$ $\boxed{\text{X-CAL}}$ $\boxed{\int dx}$ \blacktriangleright</p>	<p>$\boxed{\text{F1}}$ {ROOT} → Racines ($f(x) = 0$)</p> <p>$\boxed{\text{F2}}$ / $\boxed{\text{F3}}$ {MAX/MIN} → Maximum/Minimum</p> <p>$\boxed{\text{F4}}$ {Y-ICEPT} → Accéder au point de la courbe d'abscisse 0</p> <p>$\boxed{\text{F5}}$ {INTSECT} → Intersection entre deux courbes</p> <p>$\boxed{\text{F6}}$ $\boxed{\text{F1}}$ / $\boxed{\text{F2}}$ {Y-CAL / X-CAL} → Accéder au(x) point(s) de la courbe d'abscisse x donnée / d'ordonnée y donnée</p> <p>$\boxed{\text{F6}}$ $\boxed{\text{F3}}$ {$\int dx$} → Représenter et calculer une intégrale</p> <p>\blacktriangleleft \blacktriangleright permettent d'aller d'un point à un autre s'il y a plusieurs points donnés par G-Solv</p>
<p>(Sketch)</p> <p>$\boxed{\text{Cls}}$ $\boxed{\text{Tangent}}$ $\boxed{\text{Norm}}$ $\boxed{\text{Inverse}}$ \blacktriangleright</p>	<p>$\boxed{\text{F1}}$ {Cls} → Supprimer les dessins</p> <p>$\boxed{\text{F2}}$ {Tangent} → Tracer la tangente (appuyer deux fois sur $\boxed{\text{EXE}}$ pour voir l'équation de la tangente)</p>
<p>SHIFT MENU (SET UP)</p> <p>Input/Output: Math</p> <p>Draw Type : Connect</p> <p>Ineq Type : Intsect</p> <p>Graph Func : On</p> <p>Dual Screen : Off</p> <p>Simul Graph : Off</p> <p>Derivative : On</p> <p>$\boxed{\text{Math}}$ $\boxed{\text{Line}}$ \blacktriangledown</p>	<p>Derivative: On → Affichage du nombre dérivé et de l'équation de la tangente</p> <p>Grid: Line → Affichage de la grille en arrière plan</p>

Menu Exe-Mat

<p>OPTN F4</p> <p>SolveN(sin x=0)</p> <p>{-4π, -3π, -2π, -π, 0, π, \blacktriangleright}</p> <p>$\frac{d}{dx}((x-2)(x+1)) _{x=1}$</p> <p>$\boxed{\text{Solve}}$ $\boxed{d/dx}$ $\boxed{d^2/dx^2}$ $\boxed{\int dx}$ $\boxed{\text{SolveN}}$ \blacktriangleright</p>	<p>$\boxed{\text{F2}}$ {d/dx} → Calculer la dérivée d'une fonction en un point d'abscisse donnée</p> <p>$\boxed{\text{F4}}$ {$\int dx$} → Calculer une intégrale entre deux bornes.</p> <p>$\boxed{\text{F5}}$ {SolveN} → Résoudre une équation (il est possible de préciser l'intervalle: [a;b]: SolveN(Equation, inconnue, a, b)</p>
---	--

TABLEUR



1. Présentation du menu.....	3
2. Accéder au menu tableur.....	3
3. Se déplacer dans une feuille de calcul.....	4
1. Atteindre une cellule à l'aide des flèches	4
2. Atteindre une cellule en saisissant directement ses coordonnées	4
3. Atteindre une des extrémités d'une colonne	5
4. Atteindre une des extrémités d'une ligne	6
4. Sélectionner une plage de cellules	7
5. Insérer du contenu dans une cellule/trier les valeurs	8
1. Insérer une chaîne de caractères dans une cellule	8
2. Insérer une valeur constante dans une cellule.....	9
3. Insérer une formule dans une cellule.....	9
4. Remplacer, modifier le contenu d'une cellule	10
5. Trier les valeurs contenues par les cellules	10
6. Effacer le contenu d'une cellule, d'une ligne, d'une colonne, d'une feuille .	11
1. Effacer le contenu d'une cellule.....	11
2. Effacer le contenu d'une colonne ou d'une ligne	12
3. Effacer le contenu d'une feuille de calcul.....	13
7. Insérer ou supprimer une ligne ou une colonne	14
1. Supprimer une ligne ou une colonne	14
2. Insérer une ligne ou une colonne	14
8. Manipuler une cellule, une plage de cellules.....	15
1. Dupliquer une cellule, une plage de cellules (copier – coller)	15
2. Déplacer une cellule, une plage de cellules (couper – coller)	16
3. Stocker une plage de données sous forme de matrice/vecteur/liste	17

9.	Enregistrer, créer, ouvrir, supprimer une feuille de calcul	18
1.	Enregistrer une feuille de calcul.....	18
2.	Créer une nouvelle feuille de calcul.....	19
3.	Ouvrir une feuille de calcul enregistrée.....	20
4.	Supprimer une feuille de calcul enregistrée.....	21
10.	Dupliquer une formule	22
1.	Référence relative	22
2.	Référence absolue (utilisation du \$)	23
11.	Fonctions somme/produit/moyenne/médiane/min/max	24
12.	Condition (fonction si)	25
13.	Mise en forme conditionnelle	26
14.	Graphique et color link.....	28
15.	Les autres types de graphiques.....	30

1. Présentation du menu

Un tableur est un programme informatique capable de manipuler des feuilles de calcul (matrices). Les tableurs sont utilisés pour effectuer des tâches variées, de l'analyse statistiques simples à la production de graphiques en passant par le traitement automatisé des données financières. Chaque feuille de calcul sur la calculatrice se compose de 26 colonnes (référéncées par des lettres de A à Z) et de 999 lignes (numérotées) permettant de travailler sur des cellules. Chaque cellule référencée par une lettre (colonne) et un nombre (ligne), peut recevoir des chaînes de caractères, des constantes ou des formules.

2. Accéder au menu Tableur

A partir du menu principal, touchez **MENU**
 Se positionner à l'aide du pavé directionnel sur l'icône Tableur (ou S-SHT) pour la mettre en surbrillance.

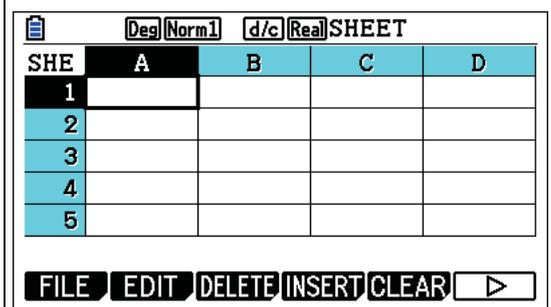
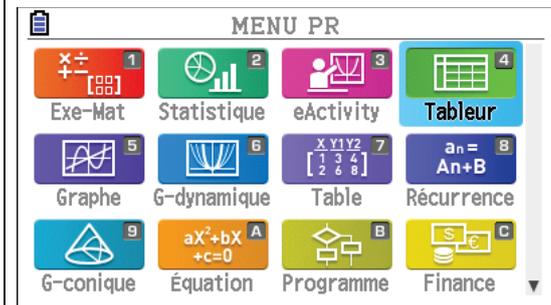
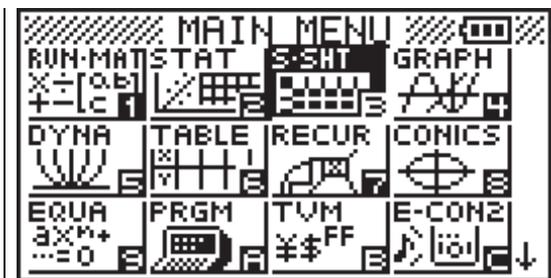


Valider à l'aide de la touche **EXE**.

Ou plus rapidement appuyer sur les touches **3** ou **4** en fonction du modèle utilisé.

La dernière feuille de calcul utilisée s'ouvre par défaut.

 Si, aucune feuille de calcul n'a été créée préalablement, une feuille vierge s'ouvre, nommée SHEET.



3. Se déplacer dans une feuille de calcul

1. Atteindre une cellule à l'aide des flèches

Objectif : Atteindre la cellule C4 à partir de la cellule A1.

 La cellule active est repérée par un contour plus épais que les autres cellules. Cette cellule est par défaut à l'ouverture d'une feuille vierge la cellule A1.

Positionner le curseur à l'aide des flèches sur la cellule C4, que l'on souhaite atteindre.

A savoir :



La cellule C4 est la cellule active.

[Deg] [Norm1] [d/c] [Real] SHEET				
SHE	A	B	C	D
1				
2				
3				
4				
5				

FILE EDIT DELETE INSERT CLEAR ▶

[Deg] [Norm1] [d/c] [Real] SHEET				
SHE	A	B	C	D
1				
2				
3				
4				
5				

FILE EDIT DELETE INSERT CLEAR ▶

2. Atteindre une cellule en saisissant directement ses coordonnées

Objectif : Atteindre la cellule E12 à partir de la cellule A1.

Le curseur est positionné sur la cellule A1.

 Cette cellule, active, est repérée par un contour plus épais que les autres cellules.

Aller dans l'onglet {EDIT} à l'aide de la touche **F2**.

[Deg] [Norm1] [d/c] [Real] SHEET				
SHE	A	B	C	D
1				
2				
3				
4				
5				

FILE **EDIT** DELETE INSERT CLEAR ▶

Aller dans l'onglet {JUMP} avec de la touche **F4**.

[Deg] [Norm1] [d/c] [Real] SHEET				
SHE	A	B	C	D
1				
2				
3				
4				
5				

CUT COPY CELL **JUMP** SEQ ▶

Appuyer sur {GO} à l'aide de la touche **F1**.

Saisir les coordonnées de la cellule à atteindre : E12.

A savoir :

ALPHA **COS** **1** **2**

Valider à l'aide de la touche **EXE**.

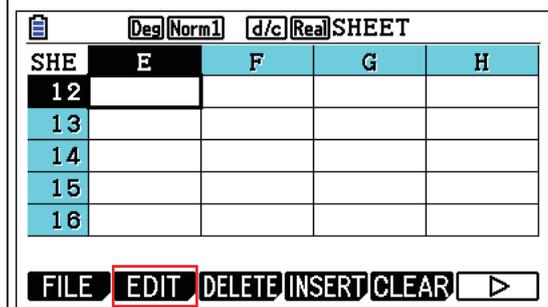
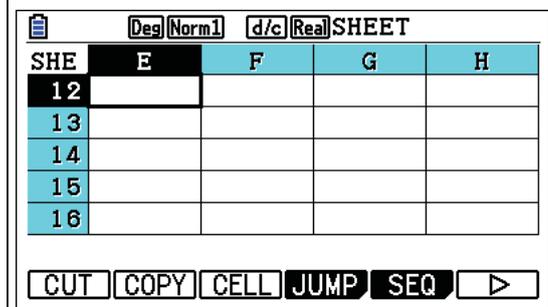
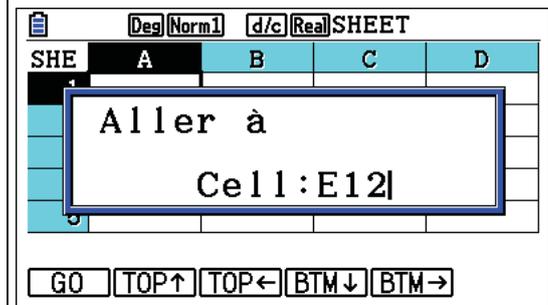
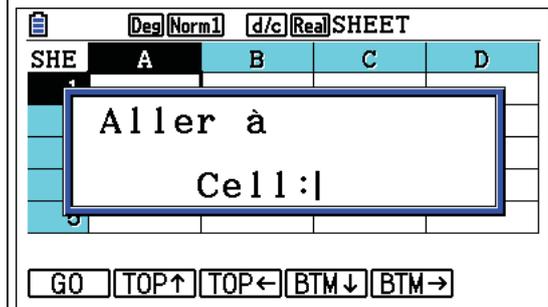
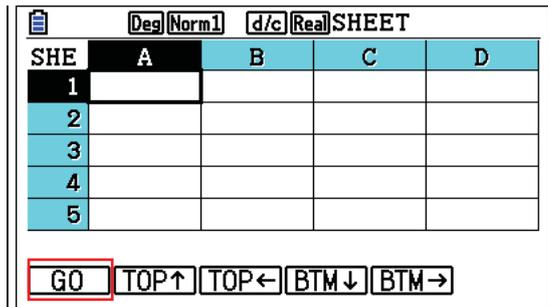
La cellule E12 est la cellule active.

3. Atteindre une des extrémités d'une colonne

Objectif : Atteindre la cellule E1 (sommet de la colonne E) à partir de la cellule E12.

La cellule E12 est active.

Aller dans l'onglet {EDIT} à l'aide de la touche **F2**.



SERIES STATISTIQUES



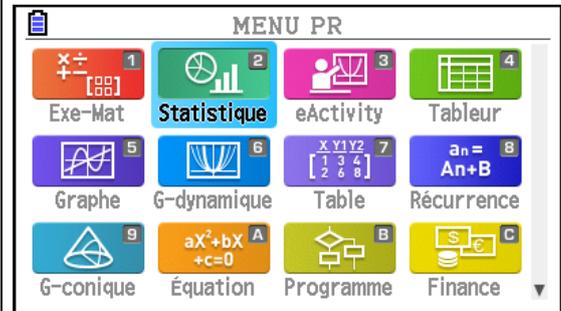
1.	Bases du menu Statistique	2
1.	Accès au menu	2
2.	Saisie des données	2
3.	Modification d'une valeur	3
4.	Insertion d'une valeur	4
5.	Suppression d'une valeur ou d'une liste spécifique	5
6.	Suppression du contenu des listes	6
7.	Nom d'une liste	7
2.	Série Statistique à une variable	8
1.	Vocabulaire et définitions	8
2.	Application	8
2.1	Fréquence de la série	9
2.2	Effectifs cumulés croissants	10
2.3	Moyenne, médiane, mode, étendue, écart-type et variance	11
2.4	Représentation graphique	14
3.	Série Statistique à deux variables	16
1.	Notion d'ajustement	16
2.	Méthode des moindres carrés	16
3.	Application	17
3.1	Coefficient de corrélation et droite de régression	18

1. Bases du menu Statistique

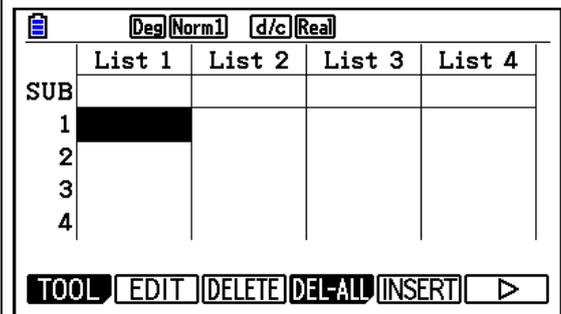
1.1. Accès au menu

À partir de la liste des menus, sélectionner à l'aide du pavé directionnel, le menu **Statistique**.
Valider en saisissant la touche **EXE**.

 Appuyer sur la touche **2** pour accéder directement au menu **Statistique** (graph 90+E), **STAT** (graph 25+E et graph 35+E).



L'éditeur de listes s'affiche. Vous pouvez ainsi saisir des données statistiques et effectuer des calculs statistiques.



1.2. Saisie des données

Application : saisir les données 10, 15, 20, 25, 30 dans la colonne "List 1".

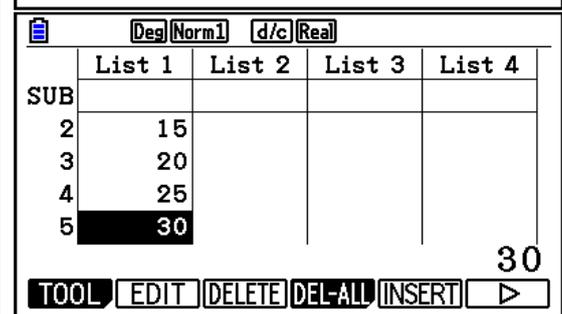
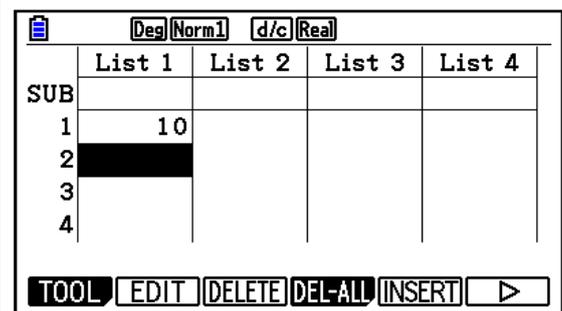
À partir du menu **Statistique**, afficher l'éditeur de listes (se référer au paragraphe 1.1).

Pour saisir les données, deux méthodes sont possibles :

- méthode 1 : saisir les valeurs une à une.

Appuyer sur les flèches **◀ ▶** pour sélectionner la colonne "List 1". Puis, saisir au clavier les différentes valeurs.

Après chaque saisie, appuyer sur la touche **EXE**.



- méthode 2 : saisir simultanément les valeurs.

Sélectionner la colonne "List 1" en utilisant le pavé directionnel.

Saisir, comme suit, les données :

{10,15,20,25,30}

Les séquences de touches **SHIFT** **X** et **SHIFT** **÷** donnent accès aux symboles { et }.

Valider avec la touche **EXE**.

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1				
2				
3				
4				
	{ 10 , 15 , 20 , 25 , 30 }			

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			
				10
TOOL EDIT DELETE DEL-ALL INSERT ▶				

1.3. Modification d'une valeur

Application : remplacer la seconde donnée "15" de la colonne "List 1" par une nouvelle donnée, par exemple "14".

Appuyer sur les flèches pour sélectionner la seconde donnée de la colonne "List 1".

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			
				15
TOOL EDIT DELETE DEL-ALL INSERT ▶				

Saisir la nouvelle valeur "14".
Valider avec la touche **EXE**.

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	14			
3	20			
4	25			
				14
TOOL EDIT DELETE DEL-ALL INSERT ▶				

1.4. Insertion d'une valeur

Application : insérer une nouvelle donnée, par exemple "16", en position 3 dans "List 1".

A l'aide du pavé directionnel, se placer en position 3 dans "List 1".

Saisir les touches **F6** {  } et **F5** {INSERT}.

Rentrer ensuite la valeur souhaitée en appuyant sur **1** **6**.

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			
				20

GRAPH CALC TEST INTR DIST 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			
				20

TOOL EDIT DELETE DEL-ALL INSERT 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	0			
4	20			
				0

TOOL EDIT DELETE DEL-ALL INSERT 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	16			
4	20			
				16

TOOL EDIT DELETE DEL-ALL INSERT 

1.5. Suppression d'une valeur ou d'une liste spécifique

Application 1 : effacer la donnée "14" présente dans "List 1".

A l'aide du pavé directionnel, se positionner sur la bonne valeur.

Presser ensuite les touches **F6** {  } et **F3** {DELETE} ({DEL} avec les graph 25+E et 35+E).

Application 2 : effacer la "List 1".

Utiliser le pavé directionnel pour sélectionner la colonne "List 1".

Presser ensuite les touches **F6** {  } et **F4** {DEL-ALL} ({DEL-A} avec les graph 25+E et 35+E).

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	14			
3	20			
4	25			

14

GRAPH CALC TEST INTR DIST 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	14			
3	20			
4	25			

14

TOOL EDIT DELETE DEL-ALL INSERT 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	20			
3	25			
4	30			

20

TOOL EDIT DELETE DEL-ALL INSERT 

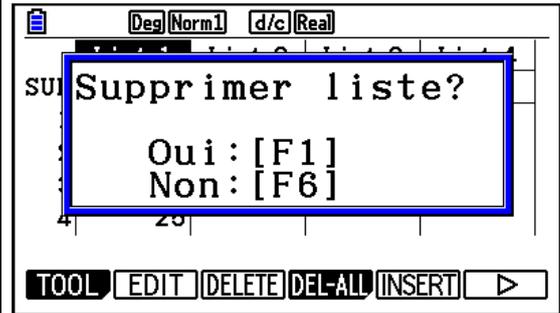
Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			

GRAPH CALC TEST INTR DIST 

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB				
1	10			
2	15			
3	20			
4	25			

TOOL EDIT DELETE DEL-ALL INSERT 

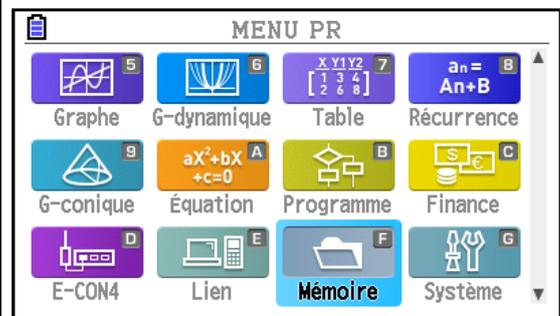
Appuyer sur la touche **[F1]** pour confirmer la suppression des données contenues dans la liste.



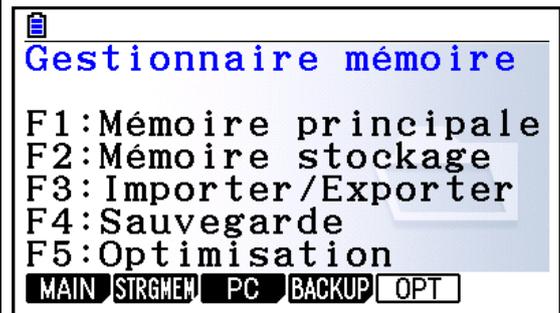
1.6. Suppression du contenu des listes

À partir de la liste de menus, sélectionner, à l'aide du pavé directionnel, le menu **Mémoire**. Valider avec la touche **[EXE]**

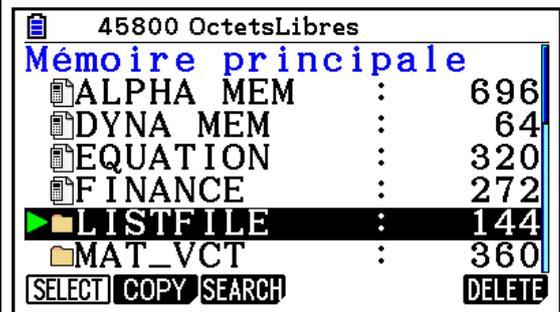
 Appuyer sur les touches **[ALPHA] [tan]** (graph 90+E), **[ALPHA] [cos]** (graph 35+E) ou **[8]** (graph 25+E) afin d'accéder directement au menu **Mémoire** (graph 90+E), **MEMORY** (graph 25+E et graph 35+E).



Le gestionnaire de mémoire s'affiche. Saisir la touche **[F1]** {Mémoire principale}.

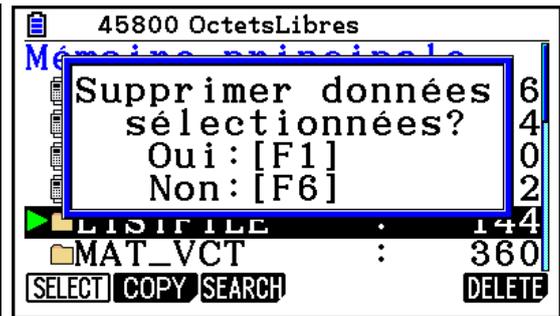


Sélectionner ensuite le dossier LISTFILE en appuyant sur **[F1]** {SELECT} ({SEL} avec les graph 25+E et 35+E). Lors que le dossier est bien sélectionné, un triangle vert apparaît :



Pour effacer l'intégralité des données statistiques, presser la touche **[F6]** {DELETE} ({DEL} avec les graph 25+E et 35+E).

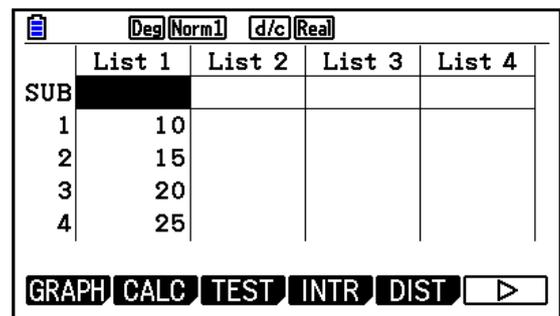
Appuyer sur la touche **[F1]** pour confirmer la suppression des données contenues dans les listes.



1.7. Nom d'une liste

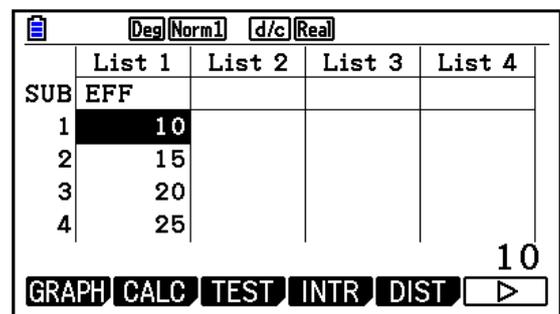
Application : nommer la colonne "List 1": EFF.

Se positionner sur la ligne [SUB] de la colonne "List 1" à l'aide du pavé directionnel.



Appuyer sur les touches **[SHIFT]** **[ALPHA]** pour verrouiller l'écriture alphabétique et accéder au mode **[ALPHA LOCK]**. Puis, saisir les lettres associées aux touches.

Saisir la touche **[SHIFT]** pour quitter ce mode
Valider en pressant la touche **[EXE]**.



2. Série Statistique à une variable

2.1. Vocabulaire et définitions

Une population est un ensemble d'individus sur lesquels on étudie un caractère ou une variable, qui prend différentes valeurs ou modalités.

Nous nous intéresserons uniquement aux variables quantitatives. Les modalités sont mesurables et prennent des valeurs numériques.

Une variable quantitative peut être :

- ✓ discrète, quand elle prend des valeurs entières,
- ✓ continue quand elle prend n'importe quelle valeur sur un intervalle donné,
- ✓ l'effectif total N : le nombre d'individus qui composent la population,
- ✓ l'effectif d'une valeur notée n_i associée à une valeur x_i est le nombre d'individus associés à cette valeur,
- ✓ la fréquence f_i : le rapport entre l'effectif de cette valeur et l'effectif total. Ainsi $f_i = \frac{n_i}{N}$,
- ✓ l'effectif (fréquence) cumulé croissant d'une valeur x_i est égal à la somme des effectifs (ou fréquences) des valeurs inférieures ou égales à x_i .

Paramètres de position :

- ✓ le mode : la ou les valeurs de la variable ayant le plus grand effectif,
- ✓ la médiane : la valeur qui partage la population en deux sous ensembles de même effectif,
- ✓ moyenne : $\bar{X} = \frac{\sum_{i=1}^n n_i x_i}{N} = \sum_{i=1}^n f_i x_i$

Paramètres de dispersions :

- ✓ l'étendue : la différence entre la plus grande valeur et la plus petite valeur de la variable,
- ✓ variance : $V(x) = \frac{\sum_{i=1}^n n_i (x_i - \bar{X})^2}{N}$
- ✓ écart type : $\sigma(x) = \sqrt{V(x)}$



N est noté n sur les calculatrices graph 25+E, graph 35+E et graph 90+E.

2.2. Application

Le comité d'entreprise d'une société propose des sorties au théâtre.

Le responsable a fait le relevé suivant pour l'année 2010.

Nombre de sorties	1	2	3	4	5	6	7
Nombre de personnes	12	17	35	29	13	9	5

- 1) Calculer les fréquences de cette série.
- 2) Calculer les effectifs cumulés croissants de cette série.
- 3) a) Calculer le mode de cette série.
b) Calculer la médiane de cette série.
c) Calculer la moyenne de cette série.
- 4) a) Calculer l'étendue de cette série.
b) Calculer la variance et l'écart type de cette série.

Saisir une à une les différentes valeurs présentées dans le tableau :

- ✓ dans "List 1" : les valeurs prises par le caractère étudié (le nombre de sorties),
- ✓ dans "List 2" : les effectifs associés (le nombre de personnes).

Nous avons aussi renommé les listes 1 (SORTIES), 2 (EFF = effectifs), 3 (F = fréquences) et 4 (ECC = effectifs cumulés croissants).

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

1

GRAPH CALC TEST INTR DIST ▶

2.2.1. Fréquence de la série

Application : calculer les fréquences de la série.

Se positionner à l'aide du pavé directionnel au niveau de la colonne "List 3".

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

GRAPH CALC TEST INTR DIST ▶

Saisir la formule permettant de calculer les fréquences :

$$(List\ 2 / \text{Sum List 2}) \times 100$$

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

(

LIST COMPLEX CALC HYPERBL PROB ▶

Pour accéder à List, appuyer sur les touches **[OPTN]**, puis **[F1]** {LIST}. Presser ensuite la touche **[F1]** {List}.

	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

(**[List]** **[List→Mat]** **[Dim]** **[Fill(]** **[Seq]** **[▶]**)

En ce qui concerne Sum : saisir les touches **[OPTN]**, et **[F1]** {LIST}. Appuyer deux fois sur **[F6]** {**[▶]**} et **[F1]** {Sum}.

	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

(**[List 2÷]** **[Sum]** **[Prod]** **[Cuml]** **[%]** **[ΔList]** **[▶]**)

Nb. sorties	1	2	3	4	5	6	7
Effectifs	12	17	35	29	13	9	5
Fréquences en %	10	14.17	29.17	24.17	10.83	7.5	4.16

	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	0	0
2	2	17		
3	3	35		
4	4	29		

2÷Sum List 2)×100|
[List] **[List→Mat]** **[Dim]** **[Fill(]** **[Seq]** **[▶]**)

Pour revenir aux onglets de base, il suffit d'appuyer sur **[EXIT]**.

	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	0
2	2	17	14.166	
3	3	35	29.166	
4	4	29	24.166	

10
[LIST] **[COMPLEX]** **[CALC]** **[HYPERBL]** **[PROB]** **[▶]**)

2.2.2. Effectifs cumulés croissants

Application : déterminer les effectifs cumulés croissants de la série.

Se positionner au niveau de la colonne "List 4".

	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	0
2	2	17	14.166	
3	3	35	29.166	
4	4	29	24.166	

[GRAPH] **[CALC]** **[TEST]** **[INTR]** **[DIST]** **[▶]**)

Saisir ensuite la formule permettant de calculer les fréquences :

Cumul List 2

Pour accéder à Cumul, presser la touche **OPTN** puis **F1** {LIST}. Presser deux fois **F6** {**▶**} et **F3** {Cuml}.

Nb. sorties	1	2	3	4	5	6	7
Effectifs	12	17	35	29	13	9	5
ECC	12	29	64	93	106	115	120

	Deg	Norm1	d/c	Real
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	0
2	2	17	14.166	
3	3	35	29.166	
4	4	29	24.166	
Cuml List 2				
List Lst→Mat Dim Fill(Seg ▶				

	Deg	Norm1	d/c	Real
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	12
2	2	17	14.166	29
3	3	35	29.166	64
4	4	29	24.166	93
				120
Sum Prod Cuml % ΔList ▶				

2.2.3. Moyenne, médiane, mode, étendue, écart-type et variance

Application : calculer les grandeurs suivantes : moyenne, médiane, mode, étendue, écart-type et variance.

Appuyer sur la touche **F2** {CALC}.

	Deg	Norm1	d/c	Real
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	12
2	2	17	14.166	29
3	3	35	29.166	64
4	4	29	24.166	93
				120
GRAPH CALC TEST INTR DIST ▶				

Presser ensuite la touche **F6** {SET} afin d'accéder aux réglages.

	Deg	Norm1	d/c	Real
	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	12
2	2	17	14.166	29
3	3	35	29.166	64
4	4	29	24.166	93
				120
1-VAR 2-VAR REG SET				

Compte tenu de notre exercice, les réglages doivent être les suivants :

- 1Var XList : List 1 (données présentes dans "List 1")
- 1Var Freq : List 2 (données présentes dans "List 2")

Se positionner à l'aide des flèches sur la seconde ligne afin de paramétrer les bons réglages. Appuyer sur la touche **F2** {LIST} puis **2**, afin de sélectionner "List 2". Valider par la touche **EXE**.

Saisir ensuite **F1** {1VAR} pour accéder aux différentes grandeurs.



	List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC
1	1	12	10	12
2	2	17	14.166	29
3	3	35	29.166	64
4	4	29	24.166	93
				1

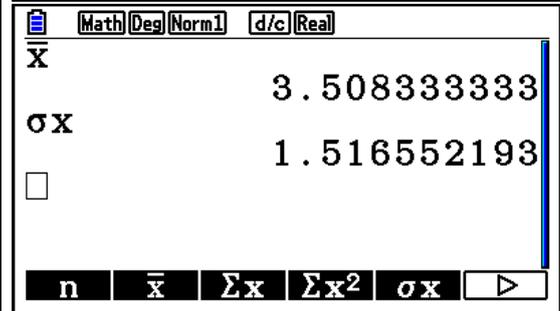
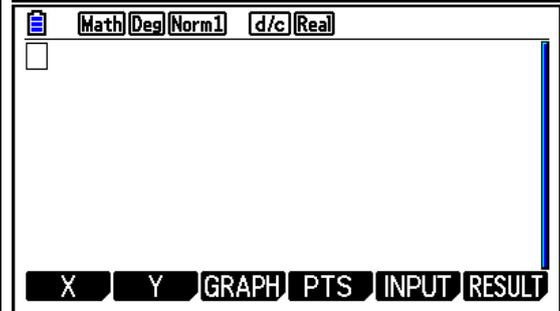
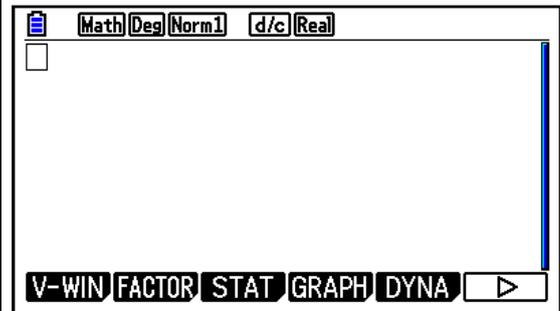
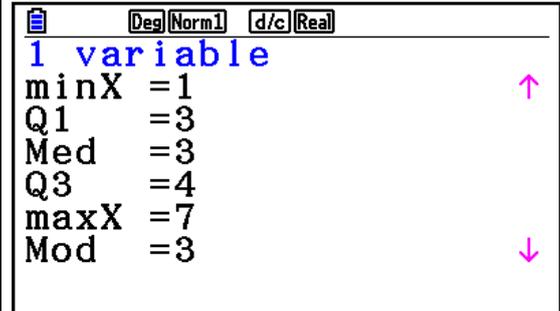
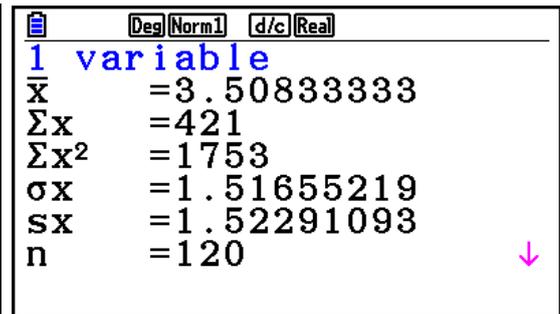
1-VAR 2-VAR REG SET

Appuyer plusieurs fois sur la touche \blacktriangledown pour faire défiler les diverses informations présentes à l'écran.

- La moyenne de cette série est d'environ 3.5.
- La médiane de cette série est de 3.
- Le mode de la série est de 3.
- L'étendue de la série est de 6.
- L'écart type de la série est d'environ 1,516.
- La variance de la série est d'environ 2,3.

Dans le menu **Exe-Mat** (graph 90+E) ou **RUN-MATH** (graph 25+E et graph 35+E), il est possible de récupérer les différentes grandeurs obtenues précédemment (moyenne, médiane...). Pour cela, appuyer sur les touches **VAR**, **F3** {STAT} et **F1** {X}.

Saisir ensuite, par exemple, **F2** { \bar{x} } ou **F5** { σx } pour obtenir la moyenne ou l'écart-type de la série.



2.2.4. Représentation graphique

Application : représenter graphiquement les effectifs (“List 2”) en fonction du nombre de sorties (“List 1”).

Presser la touche **[F1]** {GRAPH} ({GRPH} avec les graph 25+E et graph 35+E).

		List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC	
1	1	12	10	12	
2	2	17	14.166	29	
3	3	35	29.166	64	
4	4	29	24.166	93	
					1

GRAPH CALC TEST INTR DIST ▶

Appuyer sur **[F6]** {SET} afin d’accéder aux paramètres du graphique.

		List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC	
1	1	12	10	12	
2	2	17	14.166	29	
3	3	35	29.166	64	
4	4	29	24.166	93	
					1

GRAPH1 GRAPH2 GRAPH3 SELECT SET

Différents types de graphiques sont disponibles. Pour les choisir, se déplacer à l’aide du pavé directionnel sur “Graph Type”. Appuyer ensuite sur **[F1]** {Scatter} ou **[F4]** {Pie} ou **[F6]** {▶}, **[F1]** {Hist}. Valider la saisie en pressant **[EXE]**.
Le réglage Color Link permet de mettre en couleur les graphiques.

		List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC	
1	1	12	10	12	
2	2	17	14.166	29	
3	3	35	29.166	64	
4	4	29	24.166	93	
					1

StatGraph1
Graph Type : Scatter
XList : List1
YList : List2
Frequency : List2
Mark Type : ✕
Color Link : X&Y
Scatter xyLine NPPlot Pie ▶

Appuyer sur **[F1]** {GRAPH1} ({GPH1} avec les graph 25+E et graph 35+E).

		List 1	List 2	List 3	List 4
SUB	SORTIE	EFF	F	ECC	
1	1	12	10	12	
2	2	17	14.166	29	
3	3	35	29.166	64	
4	4	29	24.166	93	
					1

GRAPH1 GRAPH2 GRAPH3 SELECT SET

Nuage de points

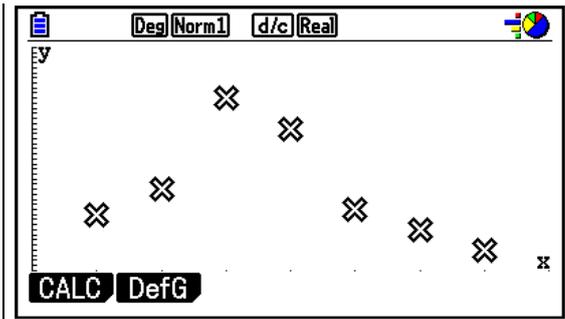
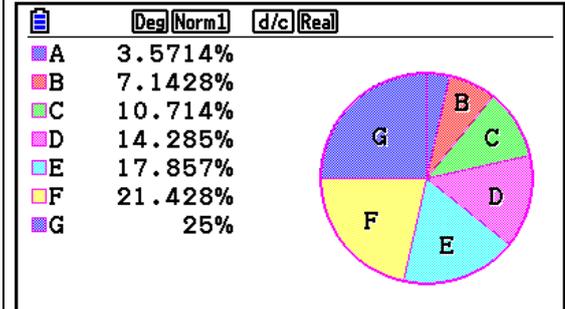
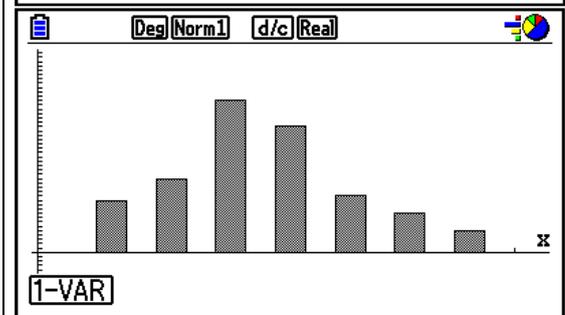


Diagramme circulaire



Histogramme



3. Série Statistique à deux variables

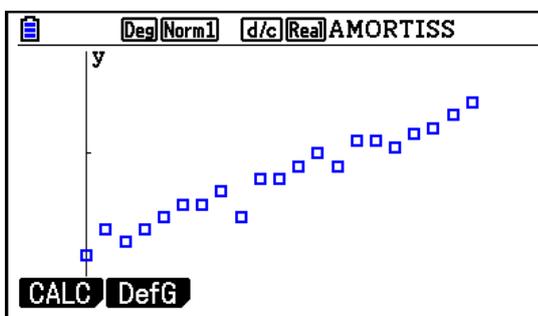
3.1. Notion d'ajustement

Quand il semble exister dans certains cas, un lien entre deux caractères x et y d'une même population, par exemple entre le poids et la taille d'un nouveau né, on les étudie simultanément en vue de faire des prévisions.

A chaque individu i correspond alors le couple $(x_i ; y_i)$ dans lequel x_i est une donnée de la variable x et y_i est une donnée de la variable y .

L'ensemble des n couples $(x_i ; y_i)$ s'appelle une série statistique à deux variables d'effectif total n .

Cette série statistique à deux variables peut être présentée sous forme de tableau ou graphiquement dans le plan muni d'un repère par le nuage des points M_i de coordonnées $(x_i ; y_i)$.



On appelle point moyen d'un nuage de n points $M_i (x_i ; y_i)$ le point G de coordonnées $(\bar{x} ; \bar{y})$ avec :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Faire un ajustement affine consiste à déterminer une droite qui passe à travers le nuage le plus près possible de chaque point.

3.2. Méthode des moindres carrés

La méthode des moindres carrés donne deux droites d'ajustement, appelées droites de régression :

- ✓ la droite de régression D de y en x a pour équation $y = ax + b$ avec :

$$a = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

✓ la droite de régression D' de x en y a pour équation $y = a'x + b'$ avec :

$$a' = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(y_i - \bar{y})^2}$$

$$b' = \bar{x} - a'\bar{y}$$

Cette méthode vise à ce que la somme des carrés de tous les écarts entre la valeur observée et la valeur estimée soit minimale.

On appelle coefficient de corrélation affine des variables x et y d'une série statistique à deux variables le nombre noté r tel que :

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Interprétation de r :

- $|r| = 1$: totale dépendance linéaire entre les 2 variables,
- $|r| = 0$: aucune dépendance linéaire entre les 2 variables,
- $0,75 < |r| < 1$: bonne corrélation.

3.3. Application

Les tailles et les poids de 10 personnes sont donnés par le tableau suivant :

Taille : x_i (cm)	174	182	170	176	171	178	173	178	186	162
Poids : y_i (kg)	71	76	65	71	68	76	62	74	84	60

Les résultats seront arrondis à 2 décimales.

- 1) Calculer la valeur du coefficient de corrélation affine entre x et y.
- 2) Déterminer une équation de la droite D, droite de régression de y en x.
- 3) Déterminer une équation de la droite D', droite de régression de x en y.

Nous avons préalablement saisi :

- ✓ dans "List 1" : les différentes valeurs correspondantes aux tailles des 10 personnes,
- ✓ dans "List 2" : les différentes valeurs correspondantes aux poids des 10 personnes.

Nous avons aussi renommé "List 1" (TAILLE) et "List 2" (POIDS).

	List 1	List 2	List 3	List 4
SUB	TAILLE	POIDS		
1	174	71		
2	182	76		
3	170	65		
4	176	71		

174

GRAPH CALC TEST INTR DIST ▶

3.3.1. Coefficient de corrélation et droite de régression

Application : déterminer le coefficient de corrélation.

Appuyer sur la touche **F2** {CALC}.

Puis, saisir la touche **F6** {SET} afin d'accéder aux réglages.

Compte tenu de notre exercice, les réglages doivent être les suivants :
 2Var XList : List 1 (données présentes dans "List 1")
 2Var Freq : List 2 (données présentes dans "List 2")

Si un réglage n'est pas bon, se positionner à l'aide du pavé directionnel sur la ligne correspondante. Appuyer ensuite sur la touche **F1** {LIST} puis sur **1** ("List 1") ou **2** ("List 2"). Valider par la touche **EXE**.

		deg	Norm1	d/c	Real
SUB	TAILLE	List 1	List 2	List 3	List 4
1	174				
2	182				
3	170				
4	176				

174

GRAPH CALC TEST INTR DIST >

		deg	Norm1	d/c	Real
SUB	TAILLE	List 1	List 2	List 3	List 4
1	174				
2	182				
3	170				
4	176				

174

1-VAR 2-VAR REG SET

		deg	Norm1	d/c	Real
1Var	XList	:List1			
1Var	Freq	:List2			
2Var	XList	:List1			
2Var	YList	:List2			
2Var	Freq	:1			

LIST

		deg	Norm1	d/c	Real
1Var	XList	:List1			
1Var	Freq	:List2			
2	Sélectionner				
2\	No. liste				
2\	List[1~26]:				

LIST

Presser ensuite les touches **F3** {REG}, **F1** {X} et **F1** {ax+b}.

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	TAILLE	POIDS		
1	174	71		
2	182	76		
3	170	65		
4	176	71		
				174
1-VAR 2-VAR REG				SET

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	TAILLE	POIDS		
1	174	71		
2	182	76		
3	170	65		
4	176	71		
				174
X	Med	X ²	X ³	X ⁴ ▶

Deg Norm1 d/c Real				
	List 1	List 2	List 3	List 4
SUB	TAILLE	POIDS		
1	174	71		
2	182	76		
3	170	65		
4	176	71		
				174
ax+b a+bx				

La valeur du coefficient de corrélation affine entre x et y est d'environ 0,92. La corrélation affine est forte.

Si l'on arrondie a et b à deux décimales, une équation de la droite de régression D est :

$$y = x - 104,73$$

D'autres régressions peuvent être utilisées, par exemple, une régression linéaire avec la méthode médiane - médiane.

Deg Norm1 d/c Real	
Régr Linéaire (ax+b)	
a	= 1.00247524
b	= -104.73316
r	= 0.92539971
r ²	= 0.85636463
MSe	= 8.51219059
y=ax+b	
COPY	

Deg Norm1 d/c Real	
Med-Med	
a	= 1.1
b	= -121.46666
y=ax+b	
COPY	

Fiche pratique: Menu PYTHON



Dans le SHELL et l'éditeur de PROGRAMMES

Ecrire les instructions lettre par lettre :

[SHIFT] [ALPHA] [A]-LOCK pour bloquer le clavier en mode alphabétique

[F5] {A ↔ a} pour passer de majuscules à minuscules ↔

1 majuscule: **[ALPHA] [A]**

Plusieurs majuscules: **[SHIFT] [ALPHA] [A]**

1 minuscule: **[ALPHA] [F5] [a]**

Plusieurs minuscules: **[SHIFT] [ALPHA] [F5] [a]**

Sélectionner toutes les commandes et symboles disponibles dans le menu **PYTHON** :

CATALOG → **[SHIFT] [4]**

Caractères (par exemple **! " # \$ % ()**)
→ **[F6] {CHAR}**

ECRAN D'ACCUEIL : liste des différents fichiers



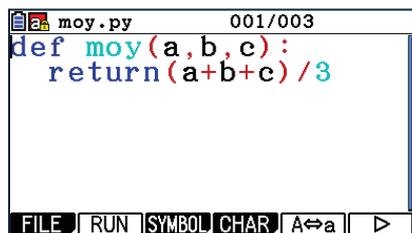
Exécuter le programme sélectionné → **[F1] {RUN}** ou **[EXE]**

Modifier le programme sélectionné → **[F2] {OPEN}**

Créer un nouveau programme → **[F3] {NEW}**

Accéder au Shell → **[F4] {SHELL}**

EDITEUR : {OPEN} ou {NEW}



Exécuter le programme → {RUN} **[F1]** ou **[EXE]**

: ; # ' " ** → {SYMBOL} **[F3]

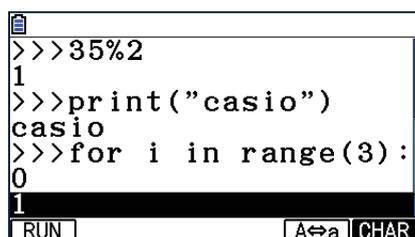
if if.else if.elif for for.range while → {COMMAND} **[F6] [F1]**

= ! > < % | ^ & ~ → {OPERAT} **[F6] [F2]**

Copier/Couper une partie d'un programme → **CLIP** **[SHIFT] [8]**
pour sélectionner une partie du programme avec puis
{COPY} **[F1]** ou {CUT} **[F2]**

Coller ce qui a été copié ou coupé → **PASTE** **[SHIFT] [9]**

SHELL : {RUN} ou {SHELL}



[F1] {RUN} ou **[EXE]** → Exécuter la ligne

Copier/Coller une ligne → remonter sur la ligne à copier () puis appuyer sur **[EXE]**

Les instructions doivent être écrites en **une seule ligne**.

LE MENU PYTHON



1.	Menu python, manipulations à partir de l'écran d'accueil et création d'un premier programme	3
1.	Accès au menu PYTHON :	3
2.	Suppression d'un fichier	3
3.	Exécution d'un programme	3
4.	Accès au contenu d'un programme existant.....	4
5.	Création et exécution d'un premier programme - Catalogue – Mode alphabétique ...	4
6.	Rechercher d'un fichier	6
2.	Le shell: l'interpréteur	6
1.	Le Copier/Coller dans le SHELL	6
2.	Calculs simples dans le SHELL, quotient, reste.....	7
3.	Les modules et les variables du menu python.....	7
1.	Le module math.....	7
2.	Nombres aléatoires, module random.....	8
3.	Les différents types de variables	8
4.	Les instructions de base: for, print, input, if	9
4.	Edition de programmes.....	11
1.	Création d'une fonction def / return , correction d'une erreur, différence entre print et return	11
2.	Instructions conditionnelles if / else / elif.....	12
3.	Boucle for	14
4.	Boucle while - module random	15
5.	Les listes:	16
6.	Programmation récursive	18
7.	Programmation avec une variable globale	20
8.	Programmation avec variables booléennes.....	21
9.	Programmation avec la bibliothèque math	22
10.	Le Copier/Coller dans l'éditeur de programmes	23
11.	Aller directement à une ligne dont le numéro est donné - JUMP:.....	23
5.	Liste des commandes les plus utiles:	24

Python est un langage de programmation interprété permettant une initiation aisée aux concepts de base de la programmation structurée. **Python** désigne également l'interpréteur qui permet de lire les scripts qui sont écrits en langage Python. La calculatrice **Graph 90+E** utilise **MicroPython** une version adaptée de **Python 3** pour les microcontrôleurs.

Nous aborderons dans ce chapitre le menu **Python** de la calculatrice graphique **CASIO Graph90+E** en nous servant de différents exercices en accord avec le programme du lycée ainsi que d'exemples traités dans le supérieur.

1. Menu Python, manipulations à partir de l'écran d'accueil et création d'un premier programme

1. Accès au menu PYTHON :

Appuyer sur la touche **MENU**
Sélectionner le menu **PYTHON** à l'aide des flèches.
Valider par la touche **EXE**

 Appuyer sur le raccourci **ALPHA** **S+D** correspondant à la lettre **{H}** en haut à droite de l'icône.

On accède ainsi à l'écran d'accueil du menu où l'on trouve l'ensemble des fichiers au format **.py** mémorisés dans la calculatrice ainsi qu'à la taille des fichiers en Ko (kilo octet).

Les dossiers **CASIO** et **SAVE-F** se créent automatiquement.

On observe la taille de chaque fichier, par exemple **AB.py** est un programme Python de taille 29 Ko.

2. Suppression d'un fichier

Application : Effacer le fichier **AB.py**

A partir de l'écran d'accueil, il est possible d'effacer un fichier en se positionnant sur le fichier concerné à l'aide des flèches **▲** **▼** et en sélectionnant **F5** **{DELETE}**.

Il faut confirmer avec la touche **F1** **{Oui}**

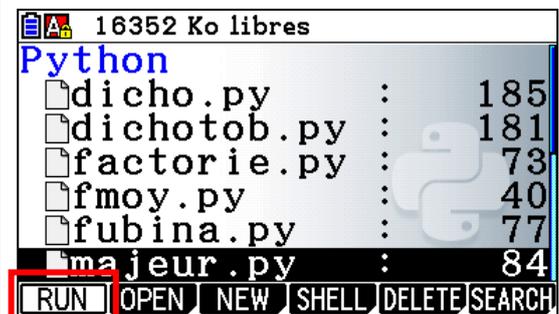
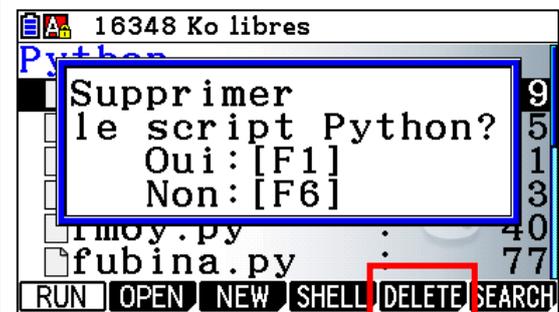
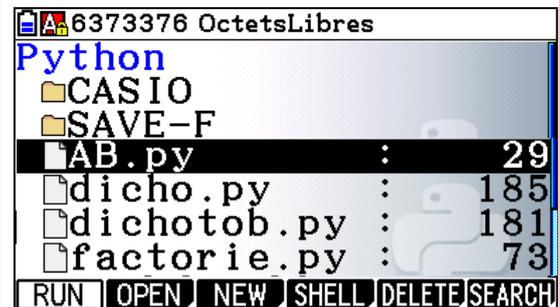
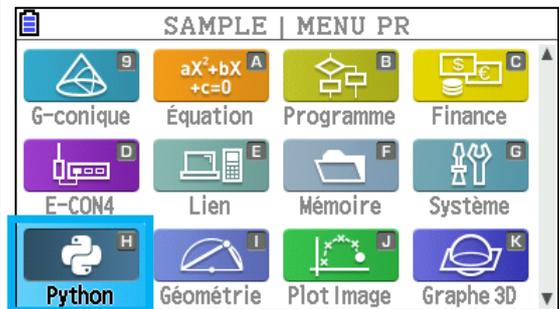
Le fichier **AB.py** est alors supprimé de la liste.

3. Exécution d'un programme

Application : Exécuter le programme **majeur.py**

A partir de l'écran d'accueil, il est possible d'exécuter un programme en se positionnant sur le fichier concerné à l'aide des flèches **▲** **▼** et en sélectionnant **F1** **{RUN}**.

Le programme est exécuté et il apparaît une ligne de commande **>>> from majeur import ***



On a ainsi téléchargé le programme **majeur** et on se retrouve dans le **SHELL** c'est à dire l'interpréteur des commandes Python.

Pour sortir du SHELL appuyer sur **EXIT**.

4. Accès au contenu d'un programme existant

Application : Accéder au contenu du programme Python *majeur.py*

A partir de l'écran d'accueil, il est possible d'accéder au contenu d'un programme en se positionnant sur le fichier concerné à l'aide des flèches **▲** **▼** et en sélectionnant **F2** **{OPEN}**.

On peut alors modifier le programme et le sauvegarder avec la touche **F1** **{FILE}** puis **F1** **{SAVE}** ou **F2** **{SAVE.AS}**. On peut aussi l'exécuter avec la touche **F2** **{RUN}**.

 On peut également sauvegarder les modifications en utilisant la touche **EXIT** et en confirmant l'enregistrement **F1** **{Oui}**.

5. Création et exécution d'un premier programme - Catalogue – Mode alphanumérique

Application : Créer un programme (script) Python appelé *carre.py* qui définit la fonction carrée.

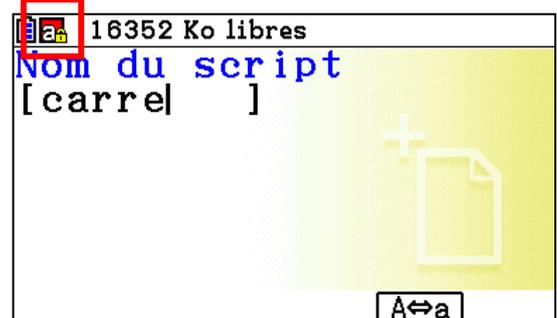
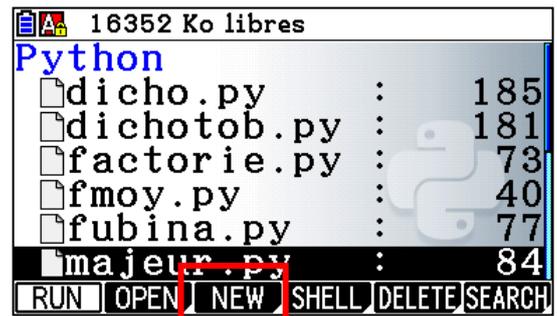
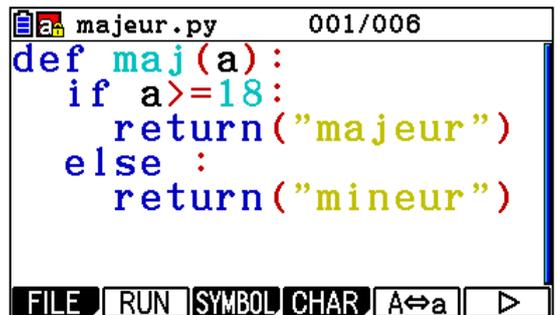
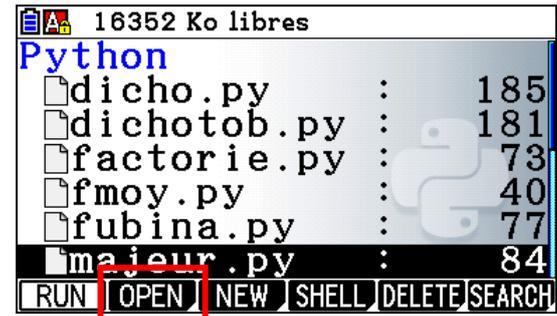
A partir de l'écran d'accueil, il est possible de créer un nouveau programme en sélectionnant **F3** **{NEW}**.

Entrer le nom du script : **carre** avec les touches alphanumériques et valider avec **EXE**.

 Par défaut, le clavier est bloqué en mode alphanumérique minuscule. Le symbole  apparait en haut à gauche de l'écran.

Si on souhaite écrire en majuscule, il d'appuyer sur la touche **F5** **{A ↔ a}**.

CATALOG:



Dans le SHELL ou l'éditeur de programme, on peut utiliser le catalogue **SHIFT** **4** (**CATALOG**) puis taper dans la barre de recherche les premières lettres de ce que l'on cherche, ici « d ». Il suffit ensuite de se déplacer à l'aide des flèches **▲** **▼** sur l'expression souhaitée et terminer par **EXE**.

On complète ensuite avec le nom de la fonction **carre** (voir le mode alphabétique ci-dessous) en précisant son argument ici « x » et le calcul à effectuer **x** **▲** **2**.

MODE ALPHABETIQUE:

Dans le SHELL ou l'éditeur de programme, les touches alphanumériques s'activent avec **ALPHA**, dans ce mode il est possible d'écrire une seule lettre.

Les touches **SHIFT** **ALPHA** permettent de bloquer le clavier en mode alphabétique, le cadenas apparaît sur l'icône. Dans ce mode il est possible d'écrire plusieurs lettres sans appuyer sur **ALPHA**.

F5 {**A** ↔ **a**} permet de passer des majuscules aux minuscules.

```

Catalogue [d ]
def
def:
def: return
del
divmod(,)
e
INPUT CAT
    
```

```

carre.py 002/003
def carre(x):
    return x**2
FILE RUN SYMBOL CHAR A↔a ▶
    
```

1 majuscule: **ALPHA** **A**

Plusieurs majuscules: **SHIFT** **ALPHA** **A**

1 minuscule: **ALPHA** **F5** **a**

Plusieurs minuscules: **SHIFT** **ALPHA** **F5** **a**

On remarque que le symbole puissance du clavier est remplacé par l'écriture de la puissance ****** en Python.

On remarque aussi la **coloration syntaxique**.

def et **return** en **bleu foncé** sont des expressions réservées à l'écriture d'instructions en Python.

On voit que le programme **carre** a été importé : **from carre import ***

Pour activer le programme dans le SHELL il faut appeler **carre** en passant une valeur en paramètre.

Par exemple, on tape dans le **SHELL** **carre(5)**

On voit alors l'affichage du résultat souhaité.

On peut recommencer avec d'autres valeurs.

On sort du SHELL avec **EXIT**.

```

* SHELL Initialized *
>>>from carre import
>>>carre(5)
25
>>>carre(9)
81
>>>|
RUN A↔a CHAR
    
```

6. Rechercher d'un fichier

Application : On souhaite trouver un programme nommé *mystere.py*

A partir de l'écran d'accueil, il est possible de chercher un fichier en sélectionnant la touche **[F6]** **{SEARCH}**.

Le nom du fichier recherché va être rentré en majuscule en validant avec **[EXE]**.

Si le fichier n'existe pas le message « **Non trouvé** »

2. Le SHELL: l'interpréteur

Le **SHELL** correspond à l'interpréteur Python.

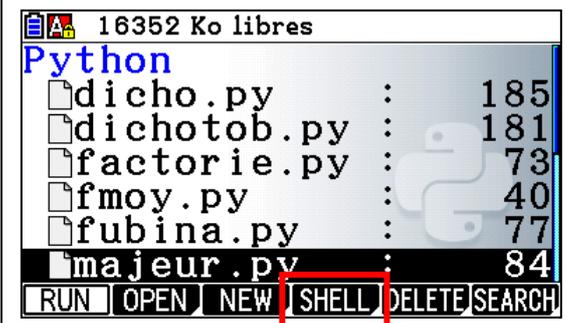
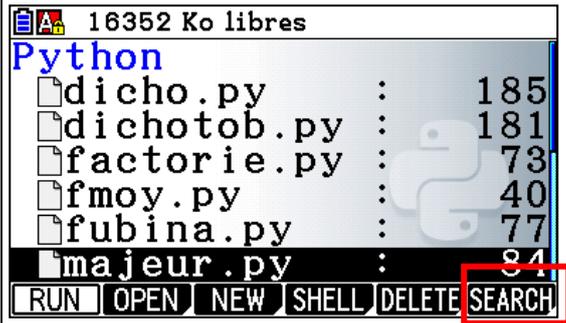
On peut exécuter des instructions Python dans le **SHELL** sans écrire un programme avec un éditeur. A partir de l'écran d'accueil, on peut ouvrir le SHELL en sélectionnant la touche **[F4]** **{SHELL}**.

On voit apparaître le message * **SHELL Initialized** * ainsi que le prompt (invite de commande) **>>>**

A partir de l'écran d'accueil, il est possible d'accéder au **SHELL** qui est un interpréteur pour exécuter des instructions Python en sélectionnant la touche **[F4]** **{SHELL}**.

1. Le Copier/Coller dans le SHELL

A l'aide des flèches **▲** **▼** on peut se placer sur une ligne du SHELL à copier puis appuyer sur **[EXE]**. Cela permet de recopier la ligne sur la dernière invite de commandes puis de la modifier.



[EXE]



[DEL] **[5]** **[EXE]**



2. Calculs simples dans le SHELL, quotient, reste

Dans le SHELL on peut effectuer toutes les opérations algébriques classiques : addition, soustraction, multiplication et division.

On peut faire d'autres calculs comme une division euclidienne chercher le quotient et le reste.

Application : On souhaite obtenir le quotient et le reste de la division de 41 par 3.

 Le quotient s'obtient à l'aide des touches   qui donnent // et le reste à l'aide de l'instruction %.

Pour afficher le symbole %, on utilise  {CHAR} et on sélectionne % avec les flèches directionnelles  puis faire .

41//3 donne un quotient de 13

41 %3 donne un reste de 3

3. Les modules et les variables du menu python

```

>>>52+3698
3750
>>>325-321
4
>>>23*654
15042
>>>|
RUN A↔a CHAR
    
```

```

|CASIO COMPUTER CO.,
* SHELL Initialized *
>>>41//3
13
>>>41%3
2
>>>|
RUN A↔a CHAR
    
```

```

Sélection Caractère
!"#$%&'()*+,-./0123
456789:;<=>?@ABCDEF
GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
    
```

Ce qui est présenté ci-dessous est aussi valable dans l'éditeur de programmes.

1. Le module math

Pour certains calculs avec des fonctions classiques comme la racine carrée ou les fonctions trigonométriques, il faut importer le module **math** avec l'instruction **from math import ***

On peut utiliser le catalogue pour écrire l'instruction avec les touches   **CATALOG** et taper dans la barre de recherche les lettres « FRO » puis se déplacer à l'aide des flèches   sur la ligne souhaitée et terminer par . Enfin, faire à nouveau  pour exécuter l'instruction.

```

Catalogue [FRO ]
from
from math import *
from random import *
getrandbits()
global
hex()
INPUT CAT
    
```

Application : On souhaite obtenir une valeur approchée à 0.001 près des réels $\sqrt{5}$ et $\sin(\frac{5\pi}{6})$.

En tapant les touches $\boxed{\text{SHIFT}} \boxed{x^2}$ on obtient l'affichage **sqrt()** qui désigne la fonction **racine carrée** en Python. Ce qui donne $\sqrt{5} \approx 2.236$

En tapant $\boxed{\sin} \boxed{(\text{C})} \boxed{5} \boxed{\times} \boxed{\text{SHIFT}} \boxed{\times 10^x} \boxed{\div} \boxed{6} \boxed{)}$

On obtient l'affichage $\sin(5 * \pi / 6)$ **pi** désigne le nombre π en Python.

On trouve alors $\sin(\frac{5\pi}{6}) = 0.5$

```
>>>from math import *
>>>sqrt(5)
2.23606797749979
>>>sin(5*pi/6)
0.4999999999999999
>>>|
RUN A↔a CHAR
```

2. Nombres aléatoires, module random

Pour générer des nombres aléatoires, il faut utiliser le module random : **from random import ***

Application : On souhaite générer un nombre entier entre 1 et 6 ainsi qu'un nombre réel compris strictement entre 0 et 1 .

Pour obtenir un nombre entier entre 1 et 6 on utilise l'instruction **randint(,)** en passant par le catalogue.

Pour obtenir un nombre réel compris strictement entre 0 et 1 on utilise l'instruction **random()**

```
>>>from random import *
>>>randint(1,6)
5
>>>randint(1,6)
4
>>>random()
0.02006387110091201
RUN A↔a CHAR
```

3. Les différents types de variables

Nous allons aborder à l'aide du **SHELL** les différents types de variables que peut gérer Python.

On dit que le langage Python a un « **typage dynamique** » car le type d'une variable est défini au moment de l'affectation.

Application : Déterminer le type des variables *a*, *b*, *c*, *d*, et *f* lorsque l'on fait les affectations suivantes :

a = 2
b = 3.14
c = "c"
d = [1,2,3]
e = (1,2,3)
f = {1,2,3}

```
>>>a=2
>>>b=3.14
>>>c="c"
>>>d=[1,2,3]
>>>e=(1,2,3)
>>>f={1,2,3}
```

 Le signe = s'obtient avec les touches:  
 Les crochets [] avec   //  
 Les accolades {} avec   //  

On effectue les affectations pour les différentes variables avec = et à l'aide de l'instruction **type()** on obtient le type de chaque variable :

- Pour a le type est « entier » (int)
- Pour b le type est « flottant » (float)
- Pour c le type est « caractère » (str)
- Pour d le type est « liste » (list)
- Pour e le type est « n-uplet » (tuple)
- Pour f le type est « ensemble » (set)

 Il existe également d'autres types comme par exemple le type booléen (bool) pour une variable qui peut prendre deux valeurs True ou False.

Si l'on fait un calcul avec une variable du type « entier » et une variable de type « flottant » et que l'on sauvegarde le résultat dans une autre variable celle-ci va avoir le type « flottant ».

Si l'on fait des opérations avec des variables du même type on conserve le type.

Par exemple, si on fait $g = c + c$ cela revient à concaténer 2 chaînes de caractères et on obtient alors que la variable g a pour contenu « cc » et qui est de type caractère « str ».

Par contre, en général on ne peut pas faire d'opérations avec des variables qui n'ont pas le même type, des messages d'erreurs s'affichent :

TypeError: unsupported types for - -add - -: 'int', 'str'
TypeError: can't convert 'list' object to str implicitly

4. Les instructions de base: for, print, input, if

Il est possible de tester une instruction dans le SHELL à condition de l'écrire sur une seule ligne. On retrouve ses instructions en sélectionnant le catalogue avec les touches   (CATALOG)

Par exemple si on veut tester la boucle **for** pour se souvenir où elle commence et où elle s'arrête, on pourra saisir :

```
>>>type(a)
<class 'int'>
>>>type(b)
<class 'float'>
>>>type(c)
<class 'str'>
```

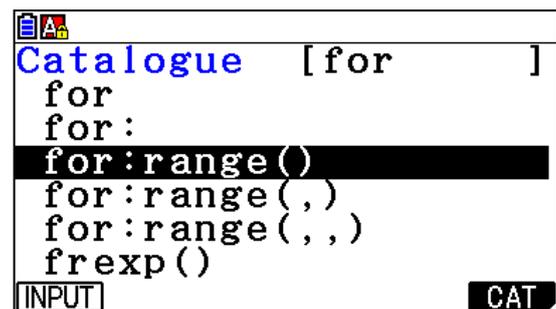
```
>>>type(d)
<class 'list'>
>>>type(e)
<class 'tuple'>
>>>type(f)
<class 'set'>
```

```
>>>g=True
>>>type(g)
<class 'bool'>
```

```
>>>g=a+b
>>>type(g)
<class 'float'>
```

```
>>>g=c+c
>>>type(g)
<class 'str'>
>>>g
'cc'
```

```
>>>g=a+c
TypeError: unsupported
>>>g=c+d
TypeError: can't conv
>>>g=d+e
TypeError: unsupporte
```



```
Catalogue [for ]
for
for:
for:range()
for:range(,)
for:range(,,)
fexp()
```

`for i in range(5) : print(i)`

Pour trouver l'instruction **for**, sélectionner le catalogue avec les touches **SHIFT** **4** (**CATALOG**) Puis taper dans la barre de recherche les lettres « FOR ». Se déplacer ensuite à l'aide des flèches **▲** **▼** sur la ligne **for : range()** et terminer par **EXE**.

Compléter la ligne avec **print(i)** que l'on trouve de la même manière dans le (**CATALOG**) puis appuyer à nouveau sur **EXE** pour valider l'instruction.

Essayons maintenant d'exécuter **n=input()**. La calculatrice se met en attente d'une saisie par l'opérateur. Si on saisit le nombre 12 on voit que la variable **n** est alors du type « **str** » et non « **int** ». L'affectation a enregistré la chaîne de caractères « 12 ».

Pour affecter un nombre entier il faut donc spécifier le type de variable en transformant la chaîne de caractère récupérée à l'aide de l'instruction **input()** en un entier. Pour cela on va utiliser l'instruction **int()** pour imposer le typage « **int** ».

L'instruction à exécuter est alors **n=int(input())** et on voit alors que pour la saisie du nombre 12 la variable **n** est bien du type « **int** ».

 On peut utiliser l'instruction **float()** pour avoir un type « flottant » et **str()** pour transformer un nombre en chaîne de caractères.

Pour réaliser des tests on fait appel aux instructions **if** ou **if else**. Les conditions du test sont par exemple de la forme **==**, **>=** ou **<=** pour comparer le contenu d'une expression avec une autre.

```
>>> range(5):print(i)
0
1
2
3
4
>>>
RUN A↔a CHAR
```

```
>>>n=input()
12
>>>type(n)
<class 'str'>
>>>n
'12'
```

```
>>>n=int(input())
12
>>>type(n)
<class 'int'>
>>>n
12
>>>|
RUN A↔a CHAR
```

```
>>>n=12
>>>float(n)
12.0
>>>str(n)
'12'
```

```
>>> if n==12:print("ok")
ok
>>>|
```

4. Edition de programmes

A la différence du SHELL, les instructions doivent ici être saisies sur plusieurs lignes. L'indentation (l'espace en début de ligne) est primordiale pour Python. Cela permet de marquer le début et la fin d'un bloc d'instructions (plus besoin de **IfEnd** / **WhileEnd** etc...). Tout ce qui est indenté par exemple après **def** fait partie de la fonction créée. L'indentation sur la calculatrice se fait automatiquement lors de la création d'une nouvelle ligne avec **[EXE]** après le symbole :

Cette indentation est par défaut égale à deux espaces mais il est possible d'en mettre

1. Création d'une fonction **def** / **return**, correction d'une erreur, différence entre **print** et **return**

Application : Nous voulons créer une fonction qui calcul la moyenne de 3 nombres réels.

On crée le programme *fmoyenne*

On utilise **[SHIFT]** **[4]** (**CATALOG**) pour aller chercher **def: return**. Le **return** est alors indenté automatiquement.

On définit la fonction nommée *fmoy* qui fait la moyenne de 3 réels *a*, *b* et *c*.

On lance le programme avec **[F2]** **{RUN}**.

Il faut confirmer l'enregistrement du programme **[F1]** **{Oui}**.

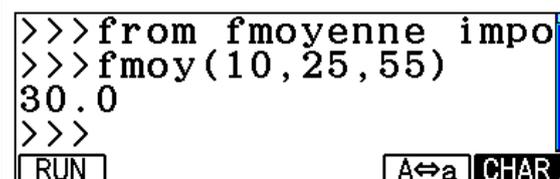
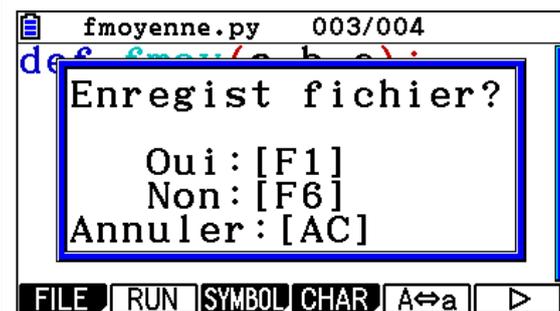
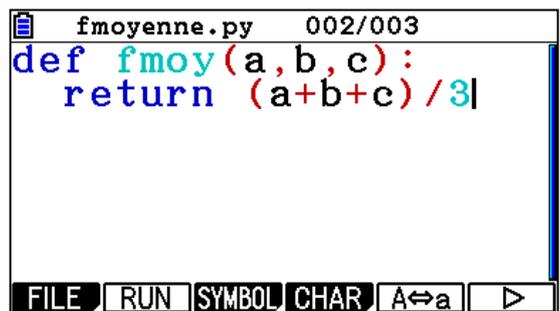
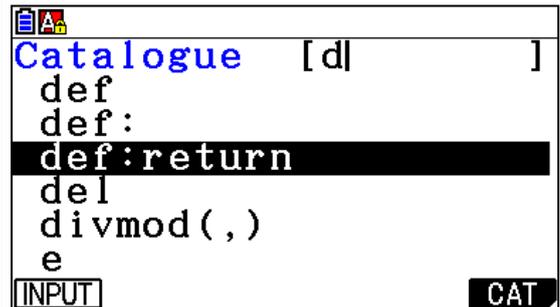
Après ce lancement on se retrouve dans le SHELL. On peut voir que l'interpréteur Python a téléchargé le nouveau programme avec l'instruction : **from fmoyenne import ***

On peut tester la fonction en tapant par exemple en ligne de commande :

>>> fmoy(10, 25, 55)

On voit que la valeur retournée par la fonction est un flottant **30.0** .

Supposons que nous ayons commis une erreur lors de l'écriture du script, par exemple, si on a oublié le symbole de la division dans le calcul de la moyenne.



Lors du lancement du programme, le **SHELL** indique qu'il y a **une erreur de syntaxe** et on peut voir que l'erreur est à la **ligne 2** du programme.

```
>>>fmoy(10,25,55)
30.0
* SHELL Initialized *
>>>from fmoyenne impo
<fmoyenne.py", line 2
SyntaxError: invalid
>>>
RUN A⇌a CHAR
```

En faisant **EXIT** on se retrouve dans l'éditeur de script et le numéro de ligne s'affiche en haut à droite.

```
fmoyenne.py 002/002
def fmoy(a,b,c):
    return (a+b+c)3
```

Si par exemple, on utilise l'instruction **print** au lieu de **return**.

```
fmoyenne.py 002/004
def fmoy(a,b,c):
    print((a+b+c)/3)
```

Lors de l'utilisation de la fonction, l'affichage sera le même mais si on fait un test du contenu de **fmoy(10,25,55)** avec la valeur **30.0**, on voit que la réponse est « fausse » (**False**).

```
>>>from fmoyenne impo
>>>fmoy(10,25,55)
30.0
>>>fmoy(10,25,55)==30
30.0
False
```

Le même test avec l'instruction **return** donne une réponse « vraie » (**True**).

En utilisant l'instruction **print** on fait juste un affichage mais on ne donne pas de valeur à **fmoy(a,b,c)** le résultat ne peut donc pas être réutilisé, par exemple **fmoy(1,2,3)+2** donnera une erreur car **fmoy(1,2,3)** sera dans ce cas un **NoneType**

```
>>>from fmoyenne impo
>>>fmoy(10,25,55)
30.0
>>>fmoy(10,25,55)==30
True
```

2. Instructions conditionnelles if / else / elif

Application : Créer une fonction qui indique si une personne est majeure ou mineure en fonction de son âge.

On crée le programme **fmajeur**.

On peut commencer ici à écrire **def** lettre par lettre (les trois lettres sont côte à côte sur le clavier), il nous faudra ici deux **return** indentés que l'on mettra après avoir inséré l'instruction conditionnelle.

On définit la fonction nommée **fmaj** qui s'applique

```
fmajeur.py 005/006
def fmaj(a):
    if a>=18:
        return("majeur")
    else:
        return("mineur")|
```

au nombre a qui est l'âge de la personne et on fait un test pour savoir si ce nombre est plus grand ou plus petit que 18.

On utilise **[SHIFT] [4] (CATALOG)** pour aller chercher **if:else** et **return** et **[F4] {CHAR}** pour sélectionner \geq .

[+/-] On peut aussi sélectionner les instructions conditionnelles avec les touches **[F6] [F1] {COMMAND}** puis **[F2] {if.else}**.

Pour revenir aux onglets principaux, appuyer sur **[EXIT] [F6]**.

On sélectionne ensuite **[F6] [F2] {OPERAT}** puis **[F3] {>} [F1] {=}** pour comparer a avec 18.

On lance le programme et on applique la fonction **fmaj** à 21 et 12. On obtient bien 'majeur' pour 21 et 'mineur' pour 12.

```
* SHELL Initialized *
>>>from fmajeur impor
>>>fmaj(21)
'majeur'
>>>fmaj(12)
'mineur'
>>>|
[RUN] A↔a CHAR
```

Application : Créer une fonction qui indique le prix unitaire d'un article en fonction de la quantité commandée N : si $0 \leq N \leq 50$ le prix sera de 15 euros, si $50 < N \leq 100$ le prix sera de 13 euros et si $N > 100$ le prix sera de 10 euros.

On crée un nouveau programme nommé **prix**.

On définit la fonction **prix** qui donne le prix unitaire de l'article en fonction de la quantité commandée N .

On sélectionne cette fois **if:elif** pour pouvoir appliquer les 3 conditions.

La fonction va retourner **10** si $N > 100$, **13** si $50 < N \leq 100$ et **15** dans les autres cas.

On peut ensuite lancer le programme et appliquer la fonction à plusieurs valeurs : 25, 60 et 150 articles.

[+/-] On pourrait ajouter autant de « elif » que l'on veut pour tester différentes possibilités.

On n'est pas obligé d'utiliser « else » ou « elif ».

```
prix.py 007/008
def prix(N):
    if N>100:
        return(10)
    elif 50<N<=100:
        return(13)
    else:
        return(15)
```

```
>>>pr ix(25)
15
>>>pr ix(60)
13
>>>pr ix(150)
10
>>>
[RUN] A↔a CHAR
```

3. Boucle for

Application : Créer une fonction qui affiche tous les diviseurs d'un entier naturel N .

On crée un nouveau programme nommé **diviseur**.

On définit la fonction **div** qui va afficher tous les diviseurs de l'entier naturel N .

On utilise **SHIFT** **4** (**CATALOG**) pour aller chercher **for:range(.)**.

⊞ On peut aussi sélectionner la boucle Pour avec les touches **F6** **F1** **{COMMAND}** puis **F4** **{for:range}**.
Pour revenir aux onglets principaux, appuyer sur **EXIT** **F6**.

L'instruction **i in range(a,b,n)** fait prendre à l'itérateur i les valeurs entières de a jusqu'à $b-1$ avec un pas de n . Si on utilise **i in range(b)** alors i prend les valeurs de 0 à $b-1$ avec un pas de 1.

On peut tester le programme par exemple avec la valeur 12.

On voit l'affichage des diviseurs de 12.

On pourrait aussi envisager de modifier le programme pour que les diviseurs de N soient sauvegardés dans une liste nommée **divis**.

Il faut initialiser la liste avec **divis= []** pour avoir une liste vide à l'initialisation.

⊞ On peut utiliser le raccourci **SHIFT** **+** **SHIFT** **=** pour saisir les crochets **[]**

On obtient la liste de tous les diviseurs de 12.

```
def div(N):
    for i in range(1,N):
        if N%i==0:
            print(i)
```

```
for i in range(1,N+1)
```

```
>>>div(12)
1
2
3
4
6
12
```

```
diviseur.py 006/007
def div(N):
    divis=[]
    for i in range(1,N+1):
        if N%i==0:
            divis=divis+[i]
    return(divis)
```

```
6
12
* SHELL Initialized *
>>>from diviseur impo
>>>div(12)
[1, 2, 3, 4, 6, 12]
>>>
```

4. Boucle while - module random

Application : Réaliser un programme qui génère aléatoirement un nombre entier entre 1 et 100. Le but sera pour l'utilisateur de trouver ce nombre. Pour l'aider, le programme lui donnera les indications « Trop petit » ou « Trop grand » à chaque fois qu'il testera un nombre. Enfin le programme donnera le score, c'est-à-dire le nombre d'essais qui auront été nécessaires à l'utilisateur pour trouver le nombre.

On crée un nouveau programme nommé **nombremy**.

On crée la fonction **mys** qui va générer un nombre aléatoire **n** que l'utilisateur va devoir trouver.

La variable **s** va compter le nombre d'essais.

La variable **a** va enregistrer le nombre choisi par l'utilisateur.

Avec **[SHIFT] [4] CATALOG** on sélectionne les différentes instructions et avec **[F4] {CHAR}** les symboles:

- La première ligne d'instructions **from random import *** permet d'importer des fonctions en lien avec les calculs de probabilités.
- La variable **s** est initialisée à 1.
- La variable **n** contient un nombre entier aléatoire entre 1 et 100 : **n=randint(1,100)**
- On demande à l'opérateur de choisir un nombre avec l'instruction **a=int(input(« nombre »))**
- On utilise une boucle while pour continuer tant que **a** est différent de **n** (**!=** est la syntaxe python pour "différent de")

[+/-] [x/y] On peut aller chercher **while** avec **[F6] [F1] {COMMAND}** puis **[F3] {while}**.

On peut ensuite sélectionner **[EXIT] [F2] {OPERAT}** puis **[F2] {!}** **[F1] {=}** ou **[F3] {>}** pour faire le test.

Pour revenir aux onglets principaux, appuyer sur **[EXIT] [F6]**.

- On incrémente le score de 1 dans chaque passage dans la boucle **s=s+1**
- On utilise l'instruction **if:else** pour afficher le message **trop grand** ou **trop petit** selon les cas.
- L'opérateur doit saisir un nouveau nombre tant que la valeur n'est pas celle de **n**.
a=int(input(« nombre »))

- Si la valeur saisie est égale à **n** alors le programme se termine et on retourne le score.
return(« bravo, score = », s)

```
from random import *
def mys():
    s=1
    n=randint(1,100)
    a=int(input("nombre
```

```
a=int(input("nombre
while a!=n:
    s=s+1
    if a>n:
        print("trop gra
    else:
        print("trop pet
```

```
a=int(input("nomb
return("bravo, scor
```

On lance le programme et on teste des valeurs pour trouver le nombre mystère.

On voit, dans la copie d'écran ci-contre, qu'au bout de 3 essais (10, 5 et 7) le nombre mystère a été trouvé.

```

nombre10
trop grand
nombre5
trop petit
nombre7
('bravo, score :', 3)
>>>
RUN A↔a CHAR
    
```

5. Les listes:

Application : Réaliser un programme qui génère **la suite de Syracuse** à partir d'un entier naturel N non nul saisi par l'utilisateur. Cette suite définie par récurrence consiste à réitérer le processus suivant :

Si le nombre est pair on le divise par 2.

Si le nombre est impair on le multiplie par 3 et on ajoute 1.

La conjecture de Syracuse, non encore démontrée à ce jour, est l'hypothèse mathématique selon laquelle l'algorithme de Syracuse appliqué à n'importe quel entier strictement positif atteint 1 au bout d'un nombre fini d'itération.

Nous allons créer un nouveau programme nommé **suitesyr**. Nous allons utiliser une liste **LIST** pour enregistrer les différentes valeurs prises par la suite jusqu'à atteindre la valeur 1.

On crée la fonction **syra** qui affiche la liste des valeurs de la suite ainsi que le nombre de termes appelé **temps de vol** de la valeur N saisie en paramètre par l'utilisateur.

```

suitesyr.py 003/006
def syra(N):
    LIST=[N]
    
```

 On peut utiliser le raccourci **SHIFT +** **SHIFT -** pour saisir les crochets **[]**

Au départ la liste **LIST** contient la valeur N.
On continue tant que N est différent de 1:
while N!=1

```

def syra(N):
    LIST=[N]
    while N!=1:
        if N%2==0:
            N=int(N/2)
        else:
            N=3*N+1
    
```

Pour faire le test de parité on utilise le reste de la division euclidienne de N par 2: **N%2**
Si celui-ci est égal à 0 c'est adire si **N%2==0** alors N est paire et sinon impair.

```

LIST=[N]
while N!=1:
    if N%2==0:
        N=int(N/2)
    else:
        N=3*N+1
LIST=LIST+[N]
    
```

 On peut utiliser le raccourcis **SHIFT**  pour saisir =
On peut trouver % dans l'onglet **{CHAR}** **F4**

Dans le cas où le nombre est pair on affecte à la variable N la valeur **int(N/2)** de type entier. Sinon on affecte à N la valeur **3*N+1**.

On ajoute le dernier élément à la liste avec

LIST=LIST+[N]

On termine par retourner la longueur `len(LIST)`
la liste `LIST`: `return(len(LIST),LIST)`

On lance le programme et on teste la fonction `syra` avec plusieurs valeurs. On voit que pour 5 la liste comporte 6 valeurs, pour 15 il y a 18 valeurs et pour 127 la liste est constituée de 47 entiers.

On remarquera que l'on peut obtenir une liste aussi grande que l'on veut avec des puissances de 2. En effet, l'entier $N = 2^p$ a pour tant de vol $p + 1$.

 Pour aller plus loin on pourrait chercher à déterminer la valeur la plus élevée atteinte par la suite, appelée altitude de la suite, en introduisant une variable `M` initialisée à `N`.

Il faut ajouter un test `if` sur la valeur de `N` par rapport à la mémoire `M`.

Si $N > M$ alors `M` prend la valeur de `N`.
A la fin de la boucle `while`, `M` contiendra l'altitude de la suite.

On constate pour une valeur initiale de 127 que l'**altitude maximale** est de **4372**.

```
turn(len(LIST),LIST)|
FILE RUN SYMBOL CHAR A↔a ▶
```

```
>>>syra(5)
◀[5, 16, 8, 4, 2, 1]
>>>syra(15)
(18, [15, 46, 23, 70,
>>>syra(127)
(47, [127, 382, 191,
>>>
RUN A↔a CHAR
```

```
def syra(N):
    LIST=[N]
    M=N
    while N!=1:
        if N%2==0:
            N=int(N/2)
```

```
        else:
            N=3*N+1
            LIST=LIST+[N]
            if N>M:
                M=N
    return(len(LIST),M,LI
```

```
>>>syra(127)
(47, [127, 382, 191,
* SHELL Initialized *
>>>from suitesyr impo
>>>syra(127)
(47, 4372, [127, 382,
>>>|
RUN A↔a CHAR
```

6. Programmation récursive

Application : Réaliser un programme qui calcule $n!$ la factorielle d'un nombre entier n :

Si $n = 0$ alors $n! = 0$

Sinon $n! = n \times (n - 1) \times \dots \times 2 \times 1 = n \times (n - 1)!$



On dit qu'un **programme est récursif** s'il s'appelle lui-même dans le programme.
On parle également de **fonction récursive**.

On crée la fonction récursive **fac** qui calcule la factorielle d'un entier naturel n entré en paramètre.

Si n est nul alors la fonction renvoie 1 sinon elle renvoie $n \times \text{fac}(n - 1)$. Cette dernière instruction de calcul fait apparaître la récursivité.

```
def fac(n):
    if n==0:
        return(1)
    else:
        return(n*fac(n-1))
```

On peut lancer le programme et tester sur plusieurs valeurs de n . on obtient, par exemples :

$3! = 3 \times 2 \times 1 = 6$ $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
 $20! = 2432902008176640000$

```
f fac(n) :
if n==0:
    return(1)
else:
    return(n*fac(n-1))
```

```
>>> fac(3)
6
>>> fac(5)
120
>>> fac(20)
2432902008176640000
```



On pourrait introduire une fonction récursive pour calculer les éléments de la suite de Syracuse abordée dans le paragraphe précédent.

```
def syra(n):
    LIST=[n]
    if n==1:
        return(LIST)
    elif n%2==0:
        return(LIST+syra(int(n/2)))
    else:
        return(LIST+syra(3*n+1))
```

```
syra.py 001/009
def syra(n) :
    LIST=[n]
    if n==1:
        return(LIST)
    elif n%2==0:
        return(LIST+syra(
    else:
```

Application : On considère la suite (u_n) définie par récurrence par $u_0 = 2$ et $u_{n+1} = 2 * u_n + 4$.

Créer un programme qui détermine le terme de rang n de la suite (u_n) .

On crée la fonction récursive **suite** qui calcule le terme de rang n . Si $n = 0$ le résultat est 2 et sinon on calcule tous les termes récursivement jusqu'à n .

```
def suite(n):
    if n==0:
        return(2)
    else:
        return(2*suite(n-1)+4)
```

```
def suite(n):
    if n==0:
        return(2)
    else:
        return(2*suite(n-1)
```

Application : Réaliser un programme récursif qui détermine à l'aide de **la méthode de dichotomie**, un encadrement avec une précision p , de l'unique solution de l'équation $f(x) = 0$ où la fonction f est continue et strictement monotone sur un intervalle $[a; b]$ avec a et b des réels tels que : $f(a) \times f(b) \leq 0$

On crée un programme nommé **dichoto** dans lequel est définie la fonction **f** suivante :

$$f(x) = x^3 + 3x - 5$$

```
def f(x):
    return (x**3+3*x-5)
```

On utilise également une fonction récursive **dicho** qui prend comme argument a , b et p .

Avec les touches une boucle **while** on fait un test sur la précision obtenue.

On utilise l'instruction **if:else** pour modifier a ou b en fonction du signe de $f(a) \times f(m)$ on fait un calcul récursif avec **dicho**.

```
def dicho(a,b,p):
    while b-a>p:
        m=(a+b)/2
        if f(a)*f(m)<=0:
            b=m
        else:
            a=m
```

Lorsque la précision est atteinte la fonction **dicho** renvoie les valeurs du dernier calcul pour a et b .

```
return(dicho(a,b,p))
return(a,b)
```

```
def dicho(a,b,p):
    while b-a>p:
        m=(a+b)/2
        if f(a)*f(m)<=0:
            b=m
        else:
            a=m
    return(dicho(a,b,p))
return(a,b)
```

On exécute alors le programme pour tester, par exemple : **dicho(1,2,0.001)** alors on obtient que la solution est entre **1.153** et **1.154**

```
>>>from dichoto impor
>>>dicho(1,2,0.001)
(1.1533203125, 1.1542
```

7. Programmation avec une variable globale

Application : On veut construire une fonction qui affecte une constante à une variable x .

La fonction f n'a pas d'argument on utilise alors des parenthèses vides $f()$

On affecte 12 à la variable x : $x = 12$

Dans le programme principal, on initialise la variable x à 0 et on utilise la fonction $f()$. Ensuite on affiche la valeur contenue dans la variable x .

Le SHELL nous affiche la valeur 0 alors que nous devrions avoir 12.

La fonction f a utilisé la variable x comme une variable locale qui n'a pas de lien avec la variable x du programme principal.

On utilise pour corrigé cela la mention **global** x à l'aide du catalogue.

La variable x devient alors globale.

C'est la même variable dans le programme principal et la fonction.

Lorsque l'on exécute le programme on obtient bien la valeur 12.

La fonction $f()$ a bien modifié le contenu de la variable x .



Par défaut une variable est toujours locale.

```
constant.py 006/006
def f():
    x=12

x=0
f()
print(x)
```

```
>>>from constant impo
0
```

```
constant.py 002/007
def f():
    global x
    x=12

x=0
f()
print(x)
```

```
* SHELL Initialized *
>>>from constant impo
12
```

8. Programmation avec variables booléennes

Application : On veut créer un script nommé **premier** qui détermine si un nombre N est un nombre premier.

On crée pour cela 3 fonctions.

La première, `divis(N,i)` renvoie **True** si le nombre i divise N et **False** sinon.

Le test qui permet de savoir si i est un diviseur de N s'écrit en python: `N%i == 0` ce qui correspond à "Le reste de la division euclidienne de N par i est égal à 0"

La deuxième fonction `nbdivis(N)` compte le nombre de diviseurs de N .

On initialise le compteur c à 0 : `c = 0`

On utilise une boucle **for** qui se répète N fois :

`for i in range(1,N+1)`

On utilise une instruction conditionnelle pour savoir si le nombre i divise N .

Si la réponse est **True** alors on incrémente le compteur c : `c = c + 1`

A la fin de la boucle la fonction va renvoyer le nombre de diviseurs soit la valeur de c .

```
def nbdivis(N):
    c=0
    for i in range (1,N+1):
        c=c+1
    return(c)
```

La dernière fonction `premier(N)` utilise un test sur le nombre de diviseur `nbdivis(N)==2` pour retourner **True** si N est premier et **False** sinon.

```
def premier(N)
    return(nbdivis(N)==2)
```

```
def divis(N, i):
    return(N%i==0)
```

```
def nbdivis(N):
    c=0
```

```
for i in range(1,N+1)
    if divis(N, i)==True:
        c=c+1
return(c)
```

```
premier(N):
return(nbdivis(N)==2)
```



Les fonctions doivent être définies avant de les utiliser dans d'autres fonctions. L'ordre de création des fonctions est donc important.

Lorsque l'on exécute le programme par exemples :

`premier(1)` donne False
`premier(17)` donne True
`premier(533)` donne False

```
>>>premier(1)
False
>>>premier(17)
True
>>>premier(533)
False
```

9. Programmation avec la bibliothèque math

Application : On veut créer un programme qui calcul la norme d'un vecteur dans l'espace muni d'un repère orthonormé.

Pour certains calculs utilisant des fonctions classiques comme la racine carrée ou les fonctions trigonométriques, il faut importer le module **math** avec l'instruction **from math import ***

On crée une fonction *norme*(*x*, *y*, *z*) où les coordonnées du vecteur dans le repère orthonormé sont des paramètres.

La fonction racine carrée en Python s'écrit **sqrt()**. Il suffit de sélectionner la touche racine carrée de la calculatrice (**SHIFT** **x²**) pour obtenir **sqrt**.

Pour la norme on rentre la formule:
 $sqrt(x^2 + y^2 + z^2)$

On remarque que le symbole puissance du clavier est remplacé par l'écriture en Python de la puissance ****** : **sqrt(x ** 2 + y ** 2 + z ** 2)**

Lorsque l'on exécute le programme par exemples :
norme(1,1,1) donne approximativement 1.732
norme(3,4,0) donne la valeur exacte 5.0

```
Catalogue [frol ]
from
from math import *
from random import *
getrandbits()
global
hex()
INPUT CAT
```

```
norme.py 004/005 ▶
from math import *
def norme(x,y,z):
    return(sqrt(x**2+y**2+z**2))
FILE RUN SYMBOL CHAR A↔a ▶
```

```
norme.py 003/004 ◀
import *
x,y,z):
qrt(x**2+y**2+z**2))|
FILE RUN SYMBOL CHAR A↔a ▶
```

```
* SHELL Initialized *
>>>from norme import
>>>norme(1,1,1)
1.732050807568877
>>>norme(3,4,0)
5.0
>>>
RUN A↔a CHAR
```

10. Le Copier/Coller dans l'éditeur de programmes

Pour Copier/Coller du texte on utilise **SHIFT** **8** **CLIP** pour sélectionner la zone à copier ensuite avec les flèches **▶**. Sélectionner ensuite **F1** **{COPY}** pour copier dans le presse papier puis déplacer le curseur à l'endroit où coller et taper **SHIFT** **9** **PASTE** pour coller le contenu du presse papier. On peut l'utiliser ici pour Copier/Coller la fonction **norme**.

11. Aller directement à une ligne dont le numéro est donné - JUMP:

Si l'on souhaite par exemple aller rapidement à un ligne donnée d'un programme, il suffit de sélectionner **F6** puis **F3** **{JUMP}**.

On a alors 3 possibilités:

- **{TOP}** pour aller à la première ligne
- **{BOTTOM}** pour aller à la dernière ligne
- **{LINE}** pour aller à une ligne de numéro donné

Dans ce dernier cas il suffit ensuite d'indiquer le numéro de la ligne souhaitée comme ci-contre.

5. Liste des commandes les plus utiles:

<code>A=3</code>	Affecte la valeur 3 à la variable A
<code>print(A)</code> <code>print("CASIO")</code>	Affiche la valeur de A Affiche le texte CASIO
<code>A=input("A=")</code> → même si une valeur est saisie, A contiendra une chaîne de caractères <code>A=int(input("A="))</code> → pour un entier <code>A=float(input("A="))</code> → pour un flottant	Demande à l'utilisateur de saisir A et affiche le texte A=
<code>A==3</code>	Teste si A est égal à 3
<code>A!=3</code>	Teste si A est différent de 3
<code>A>=3</code>	Teste si A est supérieur ou égal à 3
<code>A%3</code>	Renvoie le reste de la division de A par 3 (si $A\%3=0$ alors A est divisible par 3).

<code>def fct(a,b):</code> <code>return(résultat)</code>	Définit une fonction nommée <i>fct</i> de deux arguments <i>a</i> et <i>b</i> .
<code>if condition:</code> ... <code>else:</code> ...	Instruction conditionnelle
<code>for i in range(0,5,2):</code> ...	Pour <i>i</i> allant de 0 à 4 inclus avec un pas de 2.
<code>while condition:</code> ...	Tant que la condition est vraie répéter les instructions.

<code>l[3]</code> <code>c[3]</code>	Le 3 ^{ème} élément de la liste <i>l</i> ou de la chaîne de caractères <i>c</i> (les listes et les chaînes de caractères commencent à l'indice 0)
<code>l+[4,2]</code> <code>c+"casio"</code>	Concaténer la liste <i>l</i> avec la liste [4,2] Concaténer la chaîne <i>c</i> avec la chaîne "casio"
<code>len(l)</code> <code>len(c)</code>	Longueur d'une liste ou d'une chaîne de caractère

<code>from random import*</code>	Importe tout le module random
<code>randint(a,b)</code>	Renvoie un nombre entier aléatoire entre <i>a</i> et <i>b</i> inclus.