



Les indispensables

Algorithmique	II
1. Qu'est-ce qu'un algorithme ?	II
2. Entrées, sorties et variables	II
3. Structure conditionnelle : Si ... alors ... Sinon	IV
4. Boucle Pour	V
5. Boucle Tant Que	VI
6. Du langage naturel au langage de programmation	VII
Logiciel de géométrie dynamique	VIII
Tableur	X
Logiciel Xcas	XII
Calculatrices Casio	XIII
1. Étudier une fonction	XIV
2. Étudier une suite définie par récurrence	XV
3. Étudier une série statistique	XV
4. Effectuer une simulation d'une expérience aléatoire	XV
5. Utiliser une variable aléatoire X suivant la loi binomiale	XVI
6. Manipuler une liste définie par une formule	XVI

Algorithmique

1 Qu'est-ce qu'un algorithme ?

Un algorithme est un **ensemble d'actions à réaliser dans un ordre précis** afin d'accomplir une tâche. Le nombre d'actions à réaliser n'est pas forcément connu à l'avance mais est toujours fini.

L'élaboration d'un algorithme peut se décomposer en trois grandes phases :



Info Les algorithmes ne sont pas en lien direct avec l'informatique. Leur finalité est de fournir une réponse à un problème posé. Étant constitués d'un nombre parfois très grand d'instructions simples, leur exécution peut être laborieuse. C'est pourquoi on a recours à un ordinateur ou une calculatrice. On traduit l'algorithme dans un langage de programmation compris par la machine : on obtient alors un **programme informatique**.

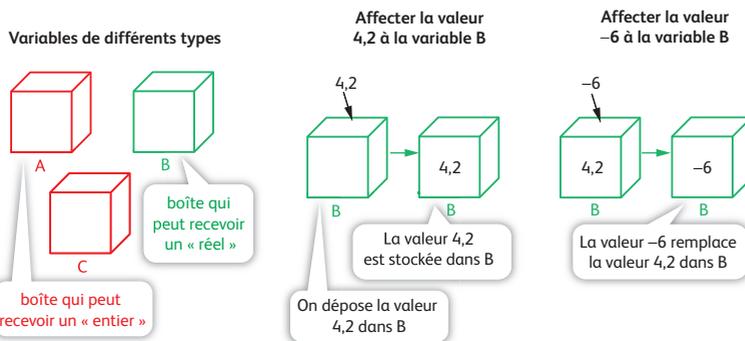
2 Entrées, sorties et variables

a Les variables

Pour pouvoir utiliser les données fournies par l'utilisateur, l'ordinateur doit les stocker dans des **variables**.

Une **variable** peut être vue comme une boîte à laquelle on donne un nom (généralement constitué d'une lettre) et qui ne peut contenir qu'une seule donnée à la fois. L'action de stocker une donnée dans une variable s'appelle l'**affectation**.

Info Attention ! Si une variable contient déjà une donnée et qu'elle est de nouveau affectée, la nouvelle donnée remplace l'ancienne qui est irrémédiablement perdue.



Lorsqu'on élabore un algorithme, il est d'usage d'indiquer l'ensemble des variables utilisées dans cet algorithme en précisant pour chacune d'elle leur **type** (entier naturel, entier relatif, nombre réel, chaîne de caractère...). On parle de **déclaration des variables**.

Remarque Dans le langage Algobox, on déclare les variables en début d'algorithme. Dans le langage des calculatrices, il n'est pas nécessaire de déclarer les variables.

Algorithmique

b Les instructions d'entrées/sorties et d'affectation

Entrées

« Saisir A » ou « Demander A » sont les instructions permettant à l'algorithme de demander à l'utilisateur de lui fournir une donnée, celle-ci étant affectée à la variable A.

Sorties

« Afficher A » est l'instruction permettant d'afficher la donnée contenue dans la variable A à l'écran.

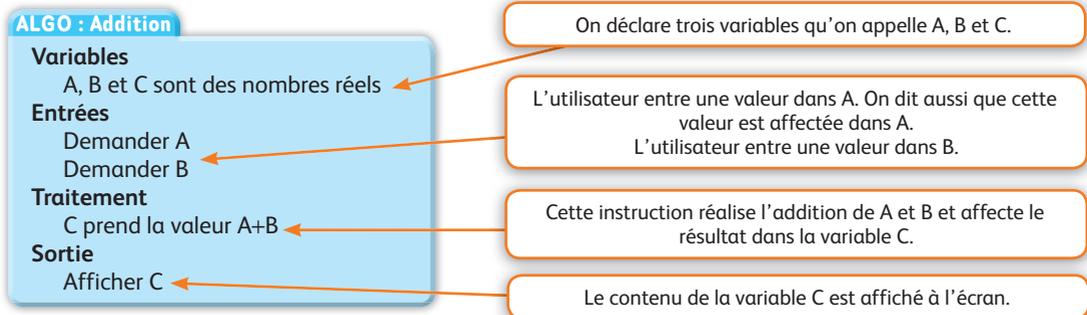
Affectation

« A prend la valeur 2 » est l'instruction permettant d'affecter la valeur 2 à la variable A.

c Exemple

Structure d'un algorithme

On souhaite élaborer un algorithme permettant d'additionner deux nombres entrés par l'utilisateur.



Application de l'algorithme pour additionner 3 et 4

Le tableau de suivi des variables A, B et C est donné ci-dessous.

Étapes de l'algorithme	A	B	C	Affichage
L'utilisateur entre 3. Cette valeur est stockée dans A	3			
L'utilisateur entre 4. Cette valeur est stockée dans B	3	4		
C prend la valeur A+B	3	4	7	
On affiche la valeur contenue dans la variable C, c'est-à-dire 7	3	4	7	7

Programmes correspondants et exécutions

Casio

```

?→A↵
?→B↵
A+B→C↵
C↵

```

```

3↵
4↵
- Disp 7

```

TI

```

:Prompt A
:Prompt B
:A+B→C
:Disp C

```

```

PrgmADDITION
A=23
B=74
Fait

```

Algobox

```

VARIABLES
├─ A EST_DU_TYPE NOMBRE
├─ B EST_DU_TYPE NOMBRE
├─ C EST_DU_TYPE NOMBRE
└─ DEBUT_ALGORITHME
    ├─ LIRE A
    ├─ LIRE B
    ├─ C PREND_LA_VALEUR A+B
    └─ AFFICHER C
FIN_ALGORITHME

```

```

***Algorithme lancé***
Entrer A : 3
Entrer B : 4
7
***Algorithme terminé***

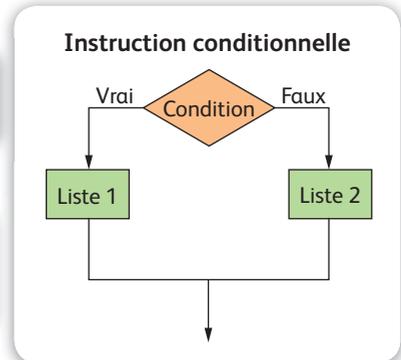
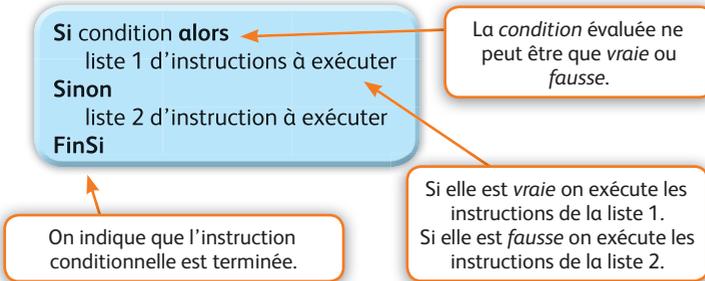
```

Algorithmique

3 Structure conditionnelle : Si... alors...Sinon...

a La structure Si... alors... Sinon...

Une instruction conditionnelle est une instruction réalisée uniquement lorsqu'une condition est remplie.



b Exemple

On souhaite élaborer un algorithme qui indique à l'utilisateur s'il est mineur ou majeur.

ALGO : Majorité

Variable : A est un nombre entier naturel
Entrée : Demander A
Traitement et sortie : Si $A < 18$ alors
 Afficher « Vous êtes mineur »
 Sinon
 Afficher « Vous êtes majeur »
 FinSi

On ne déclare ici qu'une seule variable.

Application de l'algorithme à la valeur 23

Voici le tableau de suivi des variables :

Étapes de l'algorithme	A	Condition $A < 18$	Affichage
L'utilisateur entre 23, cette valeur est stockée dans la variable A	23		
On teste la condition « $A < 18$ »	23	FAUX	
La condition n'étant pas vérifiée, on exécute les instructions du « Sinon »	23		« Vous êtes majeur »

Programmes correspondants et exécutions

Casio

```
If A<18
Then
"Vous etes mineur"
Else
"Vous etes majeur"
IfEnd
```

```
?
23
Vous etes majeur
```

TI

```
*Prompt A
*If A<18
*Then
*Disp "VOUS ETES
MINEUR"
*Else
*Disp "VOUS ETES
MAJEUR"
*End
```

```
A=23
VOUS ETES MAJEUR
Fait
```

Algobox

```
VARIABLES
| -A EST_DU_TYPE NOMBRE
|
| DEBUT_ALGORITHME
| -LIRE A
| -SI (A<18) ALORS
|   -DEBUT_SI
|   -AFFICHER "Vous êtes mineur"
|   -FIN_SI
| -SINON
|   -DEBUT_SINON
|   -AFFICHER "Vous êtes majeur"
|   -FIN_SINON
| -FIN_ALGORITHME
```

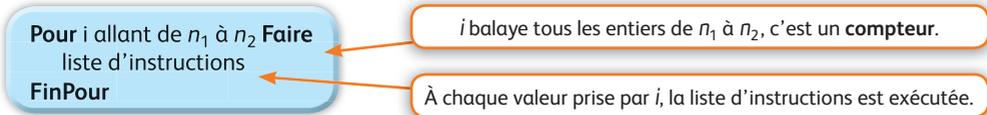
```
***Algorithme lancé***
Entrer A : 23
Vous êtes majeur
***Algorithme terminé***
```

Algorithmique

4 Boucle Pour

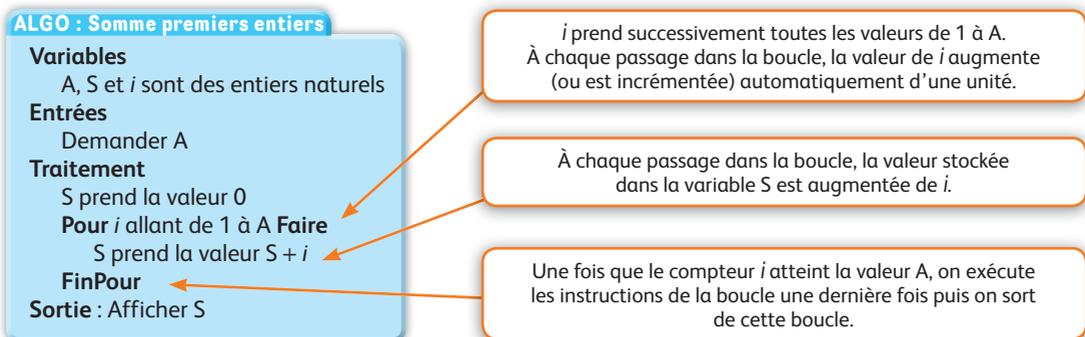
a La structure des boucles Pour

Une structure d'instructions itératives permet de répéter une liste d'instructions un nombre fini de fois ; lorsque le nombre de répétitions est connu à l'avance on dit qu'on exécute une **boucle Pour**.



b Exemple

On souhaite élaborer un algorithme qui calcule la somme de tous les entiers consécutifs de 1 à A, A étant un entier naturel entré par l'utilisateur.



Application de l'algorithme à la valeur 2

Voici le tableau de suivi des variables :

Étapes de l'algorithme	A	i	S	Affichage
L'utilisateur entre 2. Cette valeur est stockée dans la variable A. La variable S est initialisée à 0.	2		0	
1 ^{er} tour de boucle : $i = 1$, S prend la valeur $S + i$, soit $0 + 1 = 1$.	2	1	1	
2 ^e tour de boucle : $i = 2$, S prend la valeur $S + i$, soit $1 + 2 = 3$.	2	2	3	
Sortie de boucle car i a dépassé la valeur maximale (2)	2	3	3	
Le contenu de la variable S est affiché à l'écran.	2	3	3	3

Programmes correspondants et exécutions

Casio

```

?→A
0→S
For 1→I To A
S+I→S
Next
S
                    
```

TI

```

:Promt A
:0→S
:For(I,1,A)
: S+I→S
:End
:Disp S
                    
```

AlgoBox

```

VARIABLES
├─ A EST_DU_TYPE NOMBRE
├─ S EST_DU_TYPE NOMBRE
├─ I EST_DU_TYPE NOMBRE
└─ DEBUT_ALGORITHME
  ├─ LIRE A
  ├─ S PREND_LA_VALEUR 0
  └─ POUR I ALLANT DE 1 A A
    ├─ DEBUT_POUR
    │ └─ S PREND_LA_VALEUR S+I
    └─ FIN_POUR
      └─ AFFICHER S
        └─ FIN_ALGORITHME
                    
```

```

?
2
- Disp 3
                    
```

```

A=?2
3
Fait
                    
```

```

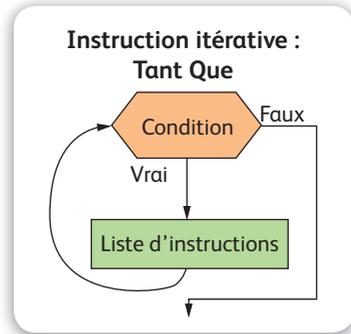
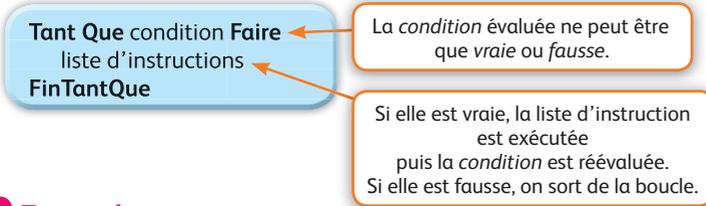
***Algorithme lancé***
Entrer A : 2
3
***Algorithme terminé***
                    
```

Algorithmique

5 Boucle Tant Que

Une structure d'instructions itératives permet de répéter une liste d'instructions un nombre fini de fois ; lorsque l'arrêt des répétitions dépend d'une condition, on dit qu'on exécute une boucle **Tant Que**.

a La structure des boucles Tant Que



b Exemple

Problème : Des parents versent la somme de 500 euros sur un compte à la naissance de leur fille. Chaque année, ils versent 200 euros sur ce compte. Combien d'années sont nécessaires pour que la somme présente sur ce compte dépasse 5 000 euros ? On suppose qu'aucun retrait n'est effectué sur ce compte.

ALGO : Évolution d'un compte bancaire

Variables : S est un nombre réel
i est un nombre entier

Entrées : S prend la valeur 500
i prend la valeur 0

Traitement : **Tant Que** $S \leq 5000$ **Faire**
S prend la valeur $S+200$
i prend la valeur $i+1$
FinTantQue

Sortie : Afficher i

S indique la somme présente sur le compte et i est un compteur qui permet de compter le nombre d'années écoulées.

À la naissance ($i = 0$), la somme présente sur le compte s'élève à 500 €.

Tant que la somme ne dépasse pas les 5 000 euros, on continue d'alimenter chaque année le compte de 200 euros et on incrémente i d'une unité. Lorsque la condition $S \leq 5000$ n'est plus vérifiée, on sort de la boucle.

Le numéro de l'année où la somme S dépasse les 5 000 € est alors stocké dans i.

Application de l'algorithme

Voici le tableau de suivi des variables :

Étapes de l'algorithme	S	i	$S \leq 5000$	Affichage
Initialisations	500	0	VRAI	
1 ^{er} tour de boucle	700	1	VRAI	
...	
22 ^e tour de boucle	4900	22	VRAI	
23 ^e tour de boucle	5100	23	FAUX	
Sortie de boucle car la condition $S \leq 5000$ n'est plus vérifiée	5100	23		
Affichage du contenu de i				23

Il faut attendre 23 années pour que la somme présente sur le compte dépasse les 5 000 euros.

Programmes correspondants et exécutions

Casio

```

=====COMPTE =====
500→S
0→I
While S≤5000
S+200→S
I+1→I
WhileEnd
    
```

- Disp 23 -

TI

```

PROGRAM:COMPTE
:500→S
:0→I
:While S≤5000
: S+200→S
: I+1→I
:End
:Disp I
    
```

23 Done

Algobox

```

VARIABLES
├── S EST_DU_TYPE NOMBRE
├── I EST_DU_TYPE NOMBRE
└── DEBUT_ALGORITHME
    ├── S PREND_LA_VALEUR 500
    ├── I PREND_LA_VALEUR 0
    └── TANT_QUE (S≤5000) FAIRE
        ├── DEBUT_TANT_QUE
        │   ├── S PREND_LA_VALEUR S+200
        │   ├── I PREND_LA_VALEUR i+1
        │   └── FIN_TANT_QUE
        └── AFFICHER i
    └── FIN_ALGORITHME
    
```

```

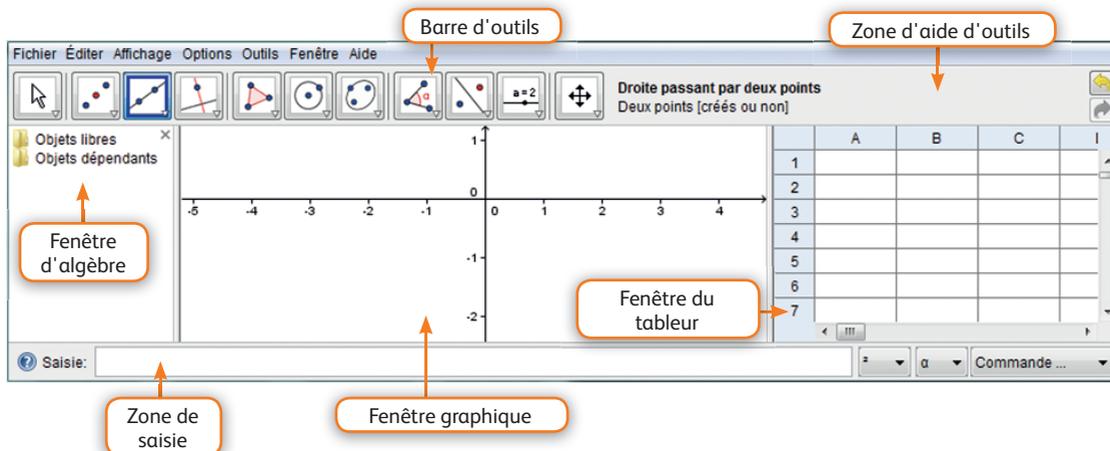
***Algorithme lancé***
23
***Algorithme terminé***
    
```

Algorithmique

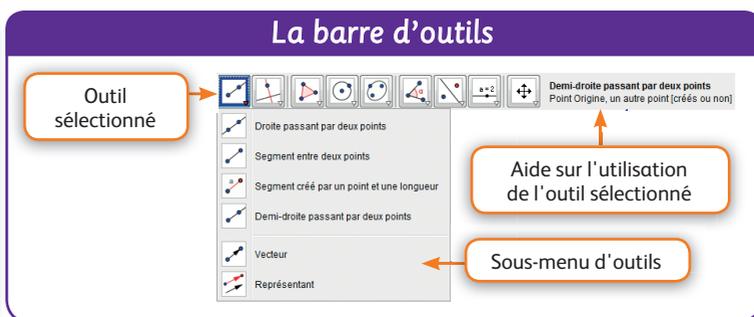
6 Du langage naturel au langage de programmation

	Casio	TI	Algobox
Manipulations spécifiques au langage	<p>On accède au mode programmation,</p> <p>MENU  E</p> <p>puis on nomme le programme :</p> <p>NEW</p> <p>Program Name []</p>	<p>On accède au mode programmation, puis on nomme le programme :</p> <p>prgm</p> <p>PROGRAMME Nom=</p>	<p>Il faut déclarer les variables utilisées :</p> <p>+ Déclarer nouvelle variable</p> <p>☒ Déclarer une variable ? ✕</p> <p>Nom de la variable : <input type="text"/></p> <p>Type de variable : NOMBRE</p> <p>OK Annuler</p> <p>Il faut ajouter une ligne pour chaque instruction :</p> <p>+ Nouvelle Ligne</p>
Entrées Demander ou Saisir	<p>Dans le menu SHIFT VARS</p> <p>On sélectionne ?</p>	<p>Dans le menu prgm (E/S)</p> <p>On sélectionne 1 :Input ou 2 :Prompt</p>	<p>+ Ajouter LIRE variable</p> <p>☒ Lire une variable ? ✕</p> <p>LIRE la variable : a</p> <p>Rang du terme de la liste :</p> <p>OK Annuler</p>
Sorties Afficher	<p>Dans le menu SHIFT VARS</p> <p>on sélectionne </p>	<p>Dans le menu prgm (E/S)</p> <p>on sélectionne 3 :Disp</p>	<p>+ Ajouter AFFICHER Variable</p> <p>AFFICHER la variable : a <input type="checkbox"/> Ajouter un retour à la ligne</p>
Affectation A prend la valeur ...	<p> A</p>	<p>Sto+ A</p>	<p>+ AFFECTER valeur à variable</p> <p>La variable : a prend la valeur : <input type="text"/></p>
Exprimer une condition	<p>Dans le menu SHIFT VARS</p> <p>on sélectionne </p> <p>puis REL</p>	<p>MENU alpha math</p>	
Instruction conditionnelle	<p>Dans le menu SHIFT VARS</p> <p>on sélectionne COM</p> <p>If Then Else End</p>	<p>Dans le menu prgm on sélectionne 1 :If 2 :Then 3 :Else et on ferme l'instruction par 7 :End</p>	<p>+ Ajouter SI...ALORS</p> <p>SI la condition : <input type="text"/> est vérifiée ALORS...</p> <p><input checked="" type="checkbox"/> Ajouter SINON</p> <p>Ne pas oublier de cocher la case « Ajouter SINON »</p>
Boucle Pour	<p>Dans le menu SHIFT VARS</p> <p>on sélectionne COM</p> <p>puis </p> <p>For To Step Next</p>	<p>Dans le menu prgm on sélectionne 4 :For on ferme la boucle 7 :End</p>	<p>+ Ajouter POUR...DE...A</p> <p>POUR la variable : a ALLANT DE <input type="text"/> A <input type="text"/></p>
Boucle Tant Que	<p>On sélectionne COM</p> <p>puis deux fois sur </p> <p>While WEnd Do LP-W</p>	<p>Dans le menu prgm on sélectionne 5 :While et on ferme la boucle par 7 :End</p>	<p>+ Ajouter TANT QUE...</p> <p>TANT QUE la condition : <input type="text"/> est vérifiée</p>

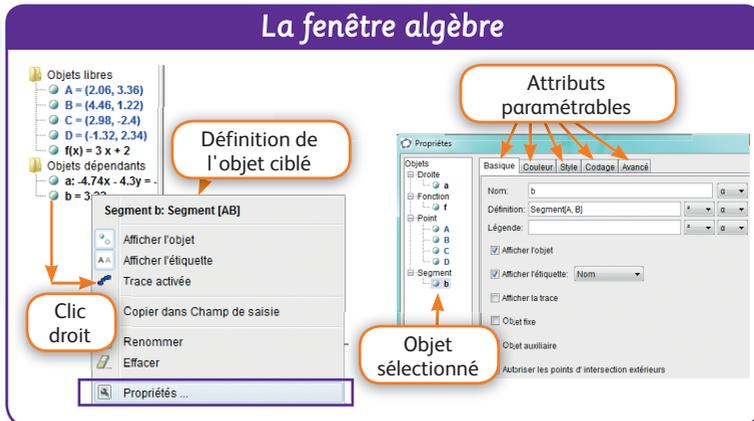
Logiciel de géométrie dynamique



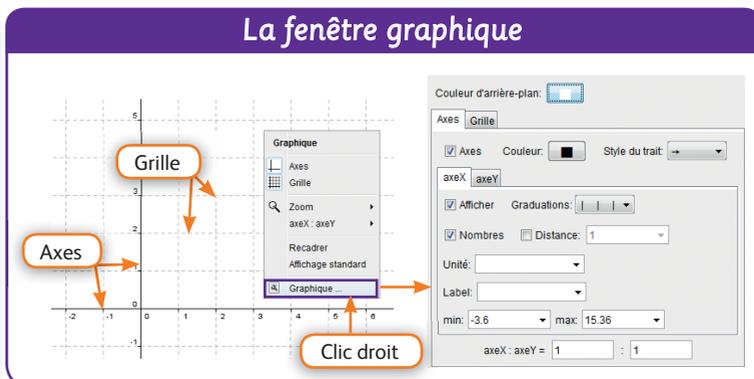
Chaque **icône** de la barre d'outils permet d'accéder à un **sous-menu** contenant des outils de même type. Lorsqu'un outil est sélectionné, une aide sur son utilisation est affichée à droite de la barre d'outils.



La fenêtre algèbre recense tous les objets créés. Elle permet d'accéder à leur **définition** et, par le biais de la fenêtre des propriétés, à l'ensemble de leurs **attributs paramétrables**.



La fenêtre graphique montre les **objets géométriques** et **mathématiques** affichables. On peut y faire apparaître un repère orthogonal et une grille dont les paramètres sont accessibles à partir du menu graphique obtenu par clic-droit dans cette fenêtre.



Logiciel de géométrie dynamique

La zone de saisie permet de **créer des objets** mathématiques à partir de commandes.

La liste de ces dernières est accessible par le menu déroulant du même nom.

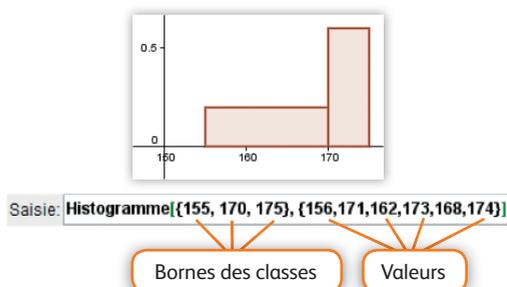
La zone de saisie

The screenshot shows the input interface with several callouts:

- Commande**: Points to the dropdown menu containing 'Extraite', 'Extremum', 'Factoriser', 'Fonction', 'Foyer', 'FractionTexte', and 'GarderSi'.
- Zone de saisie**: Points to the input field containing 'f(x)-fonctionf1/x,1,101'.
- Fonctions prédéfinies**: Points to the dropdown menu showing 'a'.
- Caractères spéciaux (dont grecs)**: Points to the 'a' dropdown.
- Liste des commandes**: Points to the 'Commande ...' dropdown.

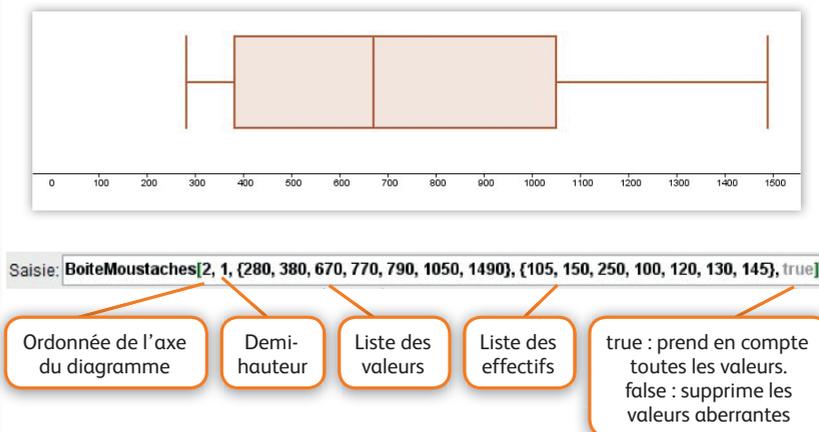
► Pour obtenir un l'histogramme de la série brute avec les classes [155 ; 170[et [170 ; 175[d'amplitudes inégales on doit directement taper la commande dans la barre de saisie.

156	162	168
171	173	174



► Pour obtenir le diagramme en boîte de la série on doit directement taper la commande dans la barre de saisie.

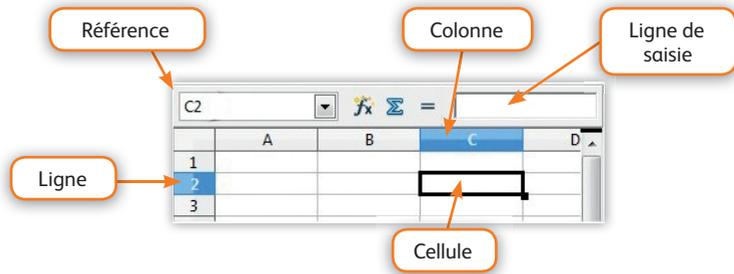
Valeurs	Effectifs
280	105
380	150
670	250
770	100
790	120
1 050	130
1 490	145



Tableur

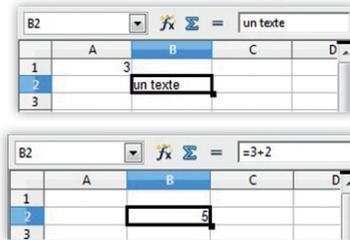
a Généralités

► Le tableur est constitué de cases appelées **cellules** et repérées par leur colonne et leur ligne. On modifie le contenu des cellules par l'intermédiaire de la **ligne de saisie**.



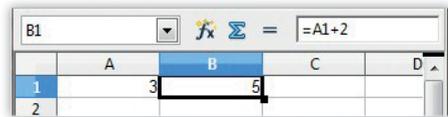
► Les cellules du tableur peuvent contenir :

- des **valeurs** (**nombre**, **texte**) et dans ce cas le contenu de la cellule et son affichage sont identiques ;
- des **formules** et dans ce cas le contenu de la cellule et son affichage sont différents.



b Formules et références

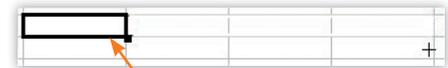
► Une formule peut utiliser le contenu d'une autre cellule en utilisant sa référence (colonne-ligne).



► Recopie vers le bas ou vers la droite

Une formule entrée dans une cellule peut être automatiquement recopiée dans les cellules situées en-dessous ou à droite grâce à la « poignée » présente dans le coin inférieur droit de la cellule sélectionnée.

En faisant un clic gauche et en maintenant, on peut étirer la sélection vers le bas ou la droite. La formule contenue dans la cellule d'origine est alors automatiquement recopiée dans toutes les cellules ainsi sélectionnées.



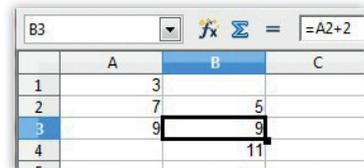
On étire la « poignée » vers la droite

Si la cellule initiale contient des références à d'autres cellules, le numéro de ligne (ou de colonne) de ces références est automatiquement augmenté.

Exemple

Si la cellule B2 contient la formule $=A1+2$ et que celle-ci est recopiée deux fois vers le bas alors :

- la cellule B3 contiendra la formule $=A2+2$
- la cellule B4 contiendra la formule $=A3+2...$



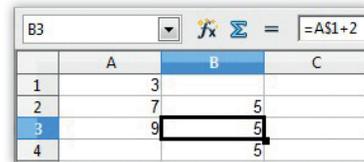
► Utilisation du signe \$

Pour éviter l'augmentation automatique de la ligne ou de la colonne lors de la recopie vers le bas ou vers la droite, on place le signe \$ devant la ligne ou la colonne (ou les deux) que l'on ne veut pas voir augmenter.

Exemple

Si la cellule B2 contient la formule $=A\$1+2$ et que celle-ci est recopiée deux fois vers le bas alors :

- la cellule B3 contiendra la formule $=A\$1+2$
- la cellule B4 contiendra la formule $=A\$1+2...$



Tableur

C Calculs

De nombreuses **fonctions mathématiques** sont utilisables dans les formules **PUISSANCE**, **RACINE**, **SOMME**...

A1	=RACINE(9)		
1	A	B	C
2			
3			

Catégorie	Mathématique
Fonction	PPCM_ADD PRODUIT PUISSANCE QUOTIENT RACINE RACINE.PI RADIAN SIGNÉ SIN SINH SOMME SOMME.CARRES SOMME.SERIES SOMME.SI SOUS.TOTAL

d Probabilités : simulation

► La fonction **ALEA()** renvoie un nombre au hasard compris entre 0 et 1 (exclu).
Lorsqu'on la recopie dans plusieurs cellules le nombre renvoyé n'est (presque) jamais le même.

A1	=ALEA()			
	A	B	C	D
1	0.46307373	0.292724609	0.199645996	0.095397949
2	Chaque cellule contient la même formule =ALEA()			
3				

► La fonction **ALEA.ENTRE.BORNES** permet d'obtenir un nombre entier entre deux bornes choisies. Elle permet par exemple de simuler le lancer d'un dé.

A1	=ALEA.ENTRE.BORNES(1;6)			
	A	B	C	D
1	5	3	4	2
2	Toutes les cellules contiennent la même formule			
3	=ALEA.ENTRE.BORNES(1;6)			
4				

► La fonction **NB.SI** permet de compter le nombre de cellules d'une plage définie respectant un certain critère.

Exemple
La formule **NB.SI** $\left[\begin{array}{l} \text{A1 : D5} \\ \text{plage} \end{array} ; \begin{array}{l} \text{" = 6"} \\ \text{critère} \end{array} \right]$ compte le nombre de cellules dans la plage A1:D5 dont le résultat est égal à 6.

F5	=NB.SI(A1:D5;"=6")							
	A	B	C	D	E	F	G	H
1	6	1	4	1				
2	4	6	2	2				
3	3	5	3	6				
4	6	2	1	6				
5	1	6	6	2		7		
6								

e Statistiques

De nombreuses **fonctions statistiques** sont disponibles afin de calculer les caractéristiques d'une série.

Si on considère la série donnée (tableau ci-contre), on peut alors en calculer les caractéristiques à l'aide des fonctions **MOYENNE**, **MEDIANE**, **QUARTILE** on peut aussi calculer les valeurs minimale et maximale de la série avec **MIN** et **MAX**.

	A	B	C	D		
1	6	1	4	1	=MOYENNE(A1:D5)	=MEDIANE(A1:D5)
2	4	6	2	2	F	G
3	3	5	3	6	Moyenne :	Médiane :
4	6	2	1	6	3,65	3,5
5	1	6	6	2		

=QUARTILE(A1:D5;1)	=QUARTILE(A1:D5;3)	=MIN(A1:D5)	=MAX(A1:D5)
F	G	F	G
Q1 :	2	Q3 :	6
		Minimum :	1
		Maximum :	6

Logiciel Xcas

1 Généralités

- ▶ On **tape les commandes** dans les lignes de calcul numérotées de l'interface. La commande saisie est exécutée lors de l'appui sur la touche « **Entrée** » du clavier.
- ▶ **Plusieurs commandes** différentes peuvent être exécutées dans une même ligne ; pour cela, elles doivent être séparées par un **point-virgule**. Les **résultats** de chaque commande sont alors affichés entre parenthèses, séparés par des virgules.
- ▶ On trouve la **totalité des commandes** existantes en parcourant les menus de l'interface (Outils, Expression, Cmds, Prg, Graphe, Geo...).

Info Le logiciel Xcas est accessible gratuitement et sans installation en ligne : http://www.xcasenligne.fr/gvac_online/demoGiac-Php.php

2 Calcul numérique

- ▶ Les résultats sont donnés de façon exacte (fractions simplifiées).
- ▶ Pour afficher une valeur approchée d'un résultat, utiliser la fonction **evalf** (en indiquant si besoin le nombre de décimales souhaitées).

Commandes particulières

- Calculer la racine carrée : **sqrt(...)**
- Déterminer la partie entière d'un nombre : **floor(...)**
- Choisir un nombre au hasard entre a et b : **rand(a,b)** ou **hasard(a,b)**
- Choisir un entier au hasard entre 1 et N : **1+rand(N)** ou **1+hasard(N)**

1	$(2+3*5) / (4*7-1)$
	$\frac{17}{27}$
2	evalf (pi, 20)
	3.1415926535897932385
3	sqrt (9); sqrt (3)
	(3, $\sqrt{3}$)
4	floor (2.6)
	(2, -4)
5	hasard (5, 9)
	7.89570647292
6	1+hasard (6)
	5

3 Calcul algébrique

Commandes particulières

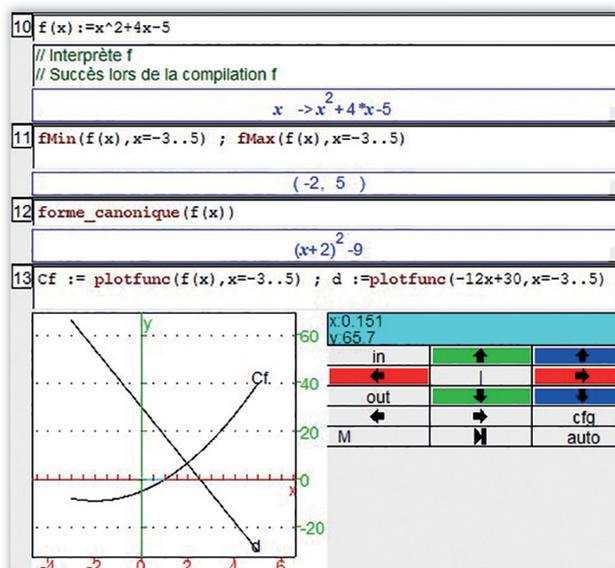
- Développer une expression : **developper(expression)**
- Factoriser une expression : **factoriser(expression)**
- Résoudre une équation : **resoudre(equation)**

7	developper ((2x+3)*(5x-2))
	$10*x^2 + 11*x - 6$
8	factoriser (3x*(5x-2)-2x*(2x+3))
	$x*(11*x-12)$
9	resoudre (2x+3=5x-2)
	$\left[\frac{5}{3} \right]$

4 Fonctions

Commandes particulières

- Définir une fonction f : **f(x) := expression**
- Déterminer la valeur en laquelle une fonction f est minimale ou maximale sur un intervalle $[a, b]$: **fMin(f(x), x=a..b)** ou **fMax(f(x), x=a..b)**
- Déterminer la forme canonique d'une fonction trinôme du second degré : **forme_canonique(f(x))**
- Tracer la courbe représentative d'une fonction f sur l'intervalle $[a, b]$: **plotfunc(f(x), x=a..b)**



Calculatrices graphiques Casio

Pour choisir le menu :



pour calculer



pour utiliser les fonctions de statistiques et de probabilités



pour tracer des courbes et utiliser le solveur graphique



pour tabuler une fonction



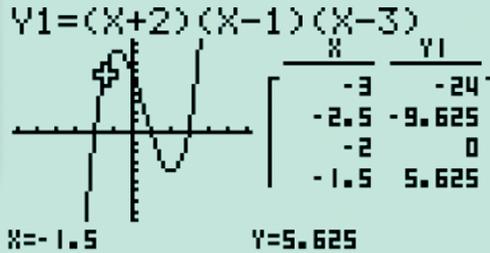
pour étudier une suite définie par récurrence



pour écrire et exécuter un programme

CASIO

GRAPH 35+



Pour choisir une commande écrite en bas de l'écran

GRAPHIQUE USB



Pour accéder à d'autres commandes, classées par grands thèmes : **LIST**, **STAT** ou **PRB**

Pour paramétrer la calculatrice : angle en degré ou radian, quadrillage affiché ou non, ...

La variable « x » lorsqu'on définit une fonction dans les menus **GRAPH** et **TABLE**

Pour élever à une puissance

Pour stocker une valeur dans une variable

Pour calculer avec des fractions

Attention à ne pas confondre ces deux touches :

- (-) pour le signe négatif d'un nombre
- pour l'opération soustraction

Calculatrices graphiques Casio

Les instructions écrites sur **fond bleu** sont affichées sur l'écran de la calculatrice, accessibles par les touches des menus **F1**, **F2**, etc.

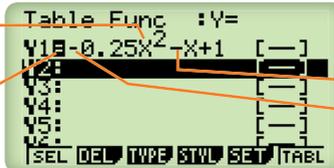
1 Étudier une fonction

On traite l'exemple de la fonction f définie par $f(x) = -0,25x^2 - x + 1$ sur $[-6 ; 3]$.

a Saisir l'expression

X est obtenu par X, θ, T

« = » est sur fond noir : l'expression est sélectionnée



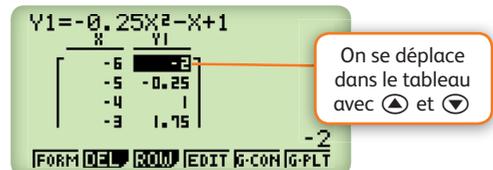
- pour l'opération soustraction
(-) pour le signe négatif de -0,25

b Dresser le tableau de valeurs

① Définir les paramètres de tabulation, en appuyant sur **SET** :

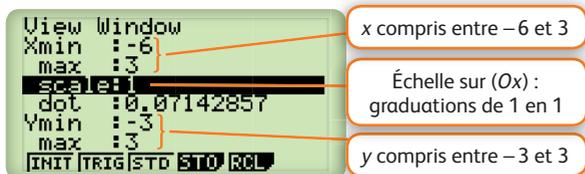


② Afficher le tableau de valeurs, en appuyant sur **TABL** :

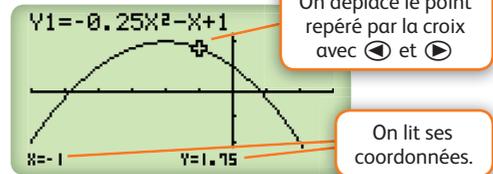


c Tracer la courbe représentative

① Définir la fenêtre d'affichage avec **SHIFT V-WIN** :

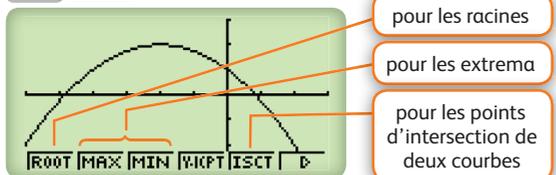


② Tracer la courbe avec **DRAW** et se déplacer avec **SHIFT TRCE** :

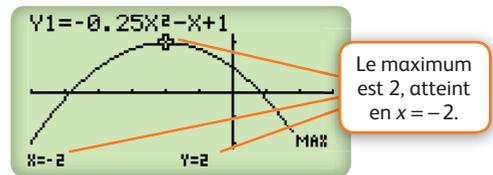


d Utiliser le solveur graphique

① Sélectionner le solveur graphique avec **SHIFT G-SLV** :



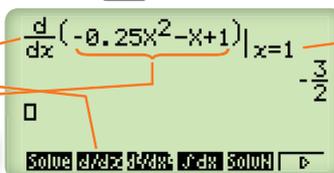
② Choisir l'instruction souhaitée et lire la solution :



e Calculer un nombre dérivé

d/dx obtenu par **OPTN CALC**

l'expression $f(x)$



le réel a

On lit $f'(a)$.
Ici, $f'(1) = \frac{3}{2}$.

Calculatrices graphiques Casio

2 Étudier une suite définie par récurrence

On traite l'exemple de la suite (u_n) définie par $u_{n+1} = 2u_n + 1$, avec $u_0 = 2$.

a Saisir l'expression

Pour avoir accès à a_n et n

```

Recursion
an+1=2an+1
bn+1:
cn+1:
    
```

Pour définir les indices des termes initial et final de la tabulation.

b Dresser le tableau de valeurs

On appuie sur **TABL** :

```

Table Settings  n+1
Start:0
End :10
a0 :2
b0 :0
c0 :0
anStr:0
a0|a1
    
```

pour définir l'indice et la valeur du terme initial.

```

an+1=2an+1
n+1  2n+1
 0  2
 1  5
 2  11
 3  23
    
```

On se déplace dans le tableau avec \uparrow et \downarrow

3 Étudier une série statistique

On traite l'exemple de la série suivante :

Valeurs x_i	3	5	8	10	13
Effectifs n_i	26	34	20	12	8

a Entrer la série

les valeurs x_i en List 1

les effectifs n_i en List 2

```

List 1 List 2 List 3 List 4
1 3 26
2 5 34
3 8 20
4 10 12
5 13 8
    
```

Pour effacer le contenu complet d'une liste.

b Calculer les paramètres statistiques

① Définir les paramètres de calculs avec **CALC SET** :

```

1Var XList :List1
1Var Freq :List2
2Var XList :List1
2Var YList :List2
2Var Freq :1
    
```

Pour choisir les listes.

② Obtenir les paramètres de la série avec **CALC 1VAR** :

```

1-Variable
x̄ =6.32
σx =6.32
σx² =4916
σx =3.03605006
sx =3.05134512
n =100
    
```

moyenne

écart type

```

1-Variable
n =100
minX =3
Q1 =8
Med =8
Q3 =8
maxX =13
    
```

médiane et quartiles

4 Effectuer une simulation d'une expérience aléatoire

Dans le **MENU**, on utilise les commandes accessibles par **OPTN RAND**.

Pour choisir au hasard un réel de $[0; 1[$.

```

Ran# 0.28095551
RanInt#(1,6) 4
RanInt#(1,6,10) {3,5,4,6,2,1,3,6,6,1}
Ran# Int Norm Bin List
    
```

Pour choisir au hasard un entier ou une liste d'entiers entre deux valeurs.

Calculatrices graphiques Casio

5 Utiliser une variable aléatoire X suivant la loi binomiale

On traite l'exemple de la loi binomiale de paramètres $n = 100$ et $p = 0,3$.

a Calculer et utiliser un coefficient binomial

① On calcule le coefficient binomial $\binom{n}{k}$ avec **OPTN** **PROB** **nCr**

② On calcule $P(X = 25) = \binom{100}{25} \times 0,3^{25} \times (1 - 0,3)^{100-25}$

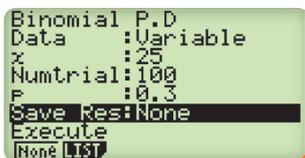
On lit $\binom{10}{4} = 210$.



On lit $P(X = 25) \approx 0,050$.

b Calculer une probabilité

① On calcule $P(X = 25)$, avec **Bpd** :



On lit $P(X \leq 25) \approx 0,050$.

Binomial P.D
P=0.04955992

② On calcule $P(X \leq 40)$, avec **Bcd** :

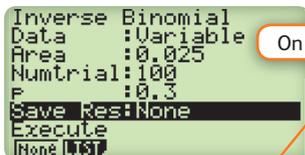


On lit $P(X \leq 40) \approx 0,988$.

Binomial C.D
P=0.98750159

c Déterminer l'intervalle de fluctuation

① On détermine le plus petit entier k_1 tel que $P(X \leq k_1) > 0,025$, avec **InvB** :



On lit $k_1 = 21$.

Inverse Binomial
xInv=21

② On détermine le plus petit entier k_2 tel que $P(X \leq k_2) \geq 0,975$, avec **InvB** :



On lit $k_2 = 39$.

Inverse Binomial
xInv=39

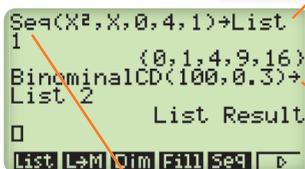
Donc $I = [0,21 ; 0,39]$.

6 Manipuler une liste définie par une formule

a Définir la liste

OPTN **LIST**

→ pour stocker dans une liste

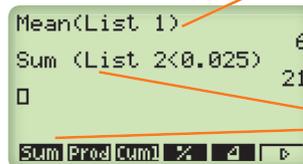


Pour définir une liste à l'aide d'une formule. Ici la somme de x^2 , pour x variant de 0 à 4, par pas de 1.

b Manipuler la liste

OPTN **LIST**

la moyenne d'une liste



le nombre de termes vérifiant la condition

On peut aussi utiliser une commande calculant une liste, ici celle des probabilités $P(X \leq k)$ où $X \sim \mathcal{B}(100 ; 0,3)$ (**OPTN** **STAT** **DIST** **BINM** **Bcd**)