

Les problèmes de syntaxe

.1 Les erreurs de syntaxe les plus courantes

- *Erreur classique n°1* : utiliser la virgule au lieu du point comme séparateur décimal
- *Erreur classique n°2* : utiliser la syntaxe "SI x=y" au lieu de "SI x==y" pour vérifier une égalité dans une condition
- *Erreur classique n°3* : utiliser la syntaxe x^y au lieu de pow(x,y) pour les puissances.

.2 Syntaxe des opérations mathématiques

Opération mathématique	syntaxe AlgoBox
\sqrt{x}	sqrt(x)
x^y	pow(x,y)
cos(x) (x en radians)	cos(x)
sin(x) (x en radians)	sin(x)
tan(x) (x en radians)	tan(x)
e^x	exp(x)
ln x	log(x)
x	abs(x)
Partie entière de x	floor(x)
Nombre pseudo-aléatoire compris entre 0 et 1	random()
Reste de la division euclidienne de n par p	n%p
π	Math.PI
arccos(x)	acos(x)
arcsin(x)	asin(x)
arctan(x)	atan(x)

Rappel : il est possible d'inclure des opérations mathématiques dans les conditions (exemple : pow(2,n)<100)

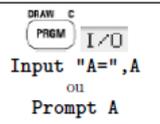
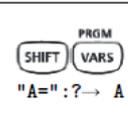
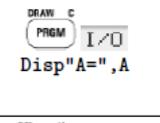
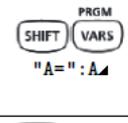
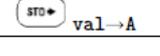
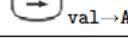
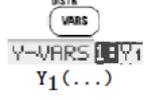
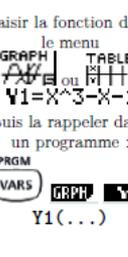
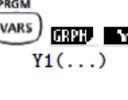
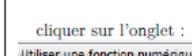
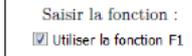
.4 Syntaxe pour les fonctions statistiques et les opérations sur les listes

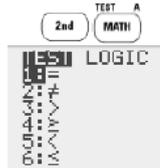
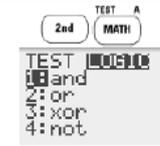
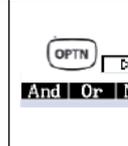
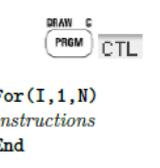
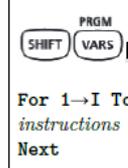
Somme des termes d'une liste	ALGOBOX_SOMME(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Moyenne	ALGOBOX_MOYENNE(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Variance	ALGOBOX_VARIANCE(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Écart-type	ALGOBOX_ECART_TYPE(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Médiane	ALGOBOX_MEDIANE(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Q1 (version calculatrice)	ALGOBOX_QUARTILE1(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Q3 (version calculatrice)	ALGOBOX_QUARTILE3(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Q1 (version « officielle »)	ALGOBOX_QUARTILE1_BIS(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Q3 (version « officielle »)	ALGOBOX_QUARTILE3_BIS(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Minimum d'une liste	ALGOBOX_MINIMUM(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Maximum d'une liste	ALGOBOX_MAXIMUM(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Rang du minimum	ALGOBOX_POS_MINIMUM(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)
Rang du maximum	ALGOBOX_POS_MAXIMUM(nom_de_la_liste,rang_premier_terme,rang_dernier_terme)

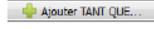
.5 Autres fonctions

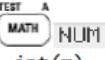
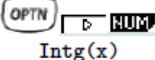
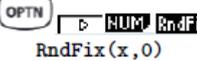
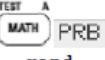
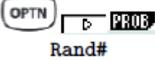
entier pseudo-aléatoire entre p et n	ALGOBOX_ALEA_ENT(p,n)
$\binom{n}{p}$ (*)	ALGOBOX_COEFF_BINOMIAL(n,p)
$p(X = k)$ pour la loi binomiale (*)	ALGOBOX_LOI_BINOMIALE(n,p,k)
$p(X < x)$ pour la loi normale centrée réduite	ALGOBOX_LOI_NORMALE_CR(x)
$p(X < x)$ pour la loi normale	ALGOBOX_LOI_NORMALE(esp,ecart,x)
x tel $p(X < x) = p$ pour la loi normale centrée réduite	ALGOBOX_INVERSE_LOI_NORMALE_CR(p)
x tel $p(X < x) = p$ pour la loi normale	ALGOBOX_INVERSE_LOI_NORMALE(esp,ecart,p)
n! (*)	ALGOBOX_FACTORIELLE(n)

(*) : avec $n < 70$

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algebox	Logiciel Xcas
Déclarer une variable A	Inutile	Inutile	➤ Déclarer nouvelle variable	<code>local A;</code>
Saisir A	 Input "A=" ,A ou Prompt A	 "A=" : ? → A	➤ Ajouter LIRE variable	<code>saisir("Entrer A" ,A);</code> ou <code>saisir(A);</code> ou si on a une fonction : <code>nom_programme(A) :={</code> <code>instruction(s); };</code>
Afficher A	 Disp "A=" ,A	 "A" : A	➤ Ajouter AFFICHER Variable	<code>afficher("A vaut :",A);</code> ou <code>afficher(A);</code> ou si on a une fonction : <code>nom_programme(paramètres) :={</code> <code>instruction(s);</code> <code>retourne A; };</code>
Affecter à A la valeur val	 val → A	 val → A	➤ AFFECTER valeur à variable	A:=val;
Utiliser une fonction externe dans un programme	Saisir la fonction dans l'éditeur graphique  puis la rappeler dans un programme :  Y1(...)	Saisir la fonction dans le menu  puis la rappeler dans un programme :  Y1(...)	cliquer sur l'onglet :  Saisir la fonction : <input checked="" type="checkbox"/> Utiliser la fonction F1 F1(x)=  puis la rappeler dans un programme : F1(...)	Définir la fonction (3 méthodes) : f(x) := x³ - x - 1 f := x → x³ - x - 1 f := unapply(x³ - x - 1, x) On peut aussi utiliser une fonction comme variable d'un programme : <code>nom_programme() :={</code> <code>local f, ...;</code> <code>saisir(f); ... };</code> Dans ce cas il faudra saisir dans l'invite : x → ...

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algebox	Logiciel Xcas
Opérateurs de test et de logique				
Opérateurs de tests $=, \neq, >, <, \geq, \leq$			<ul style="list-style-type: none"> 'x = 2' s'écrit <code>x==2</code> 'x ≠ 2' s'écrit <code>x!=2</code> 'x < 2' s'écrit <code>x<2</code> 'x > 2' s'écrit <code>x>2</code> 'x ≤ 2' s'écrit <code>x<=2</code> 'x ≥ 2' s'écrit <code>x>=2</code> 	<ul style="list-style-type: none"> 'x = 2' s'écrit <code>x==2</code> 'x ≠ 2' s'écrit <code>x!=2</code> 'x < 2' s'écrit <code>x<2</code> 'x > 2' s'écrit <code>x>2</code> 'x ≤ 2' s'écrit <code>x<=2</code> 'x ≥ 2' s'écrit <code>x>=2</code>
Opérateurs logiques et, ou, ou exclusif, non			<ul style="list-style-type: none"> le 'et' s'écrit <code>ET</code> le 'ou' s'écrit <code>OU</code> 	<ul style="list-style-type: none"> le 'et' s'écrit <code>et</code> le 'ou' s'écrit <code>ou</code> le 'ou exclusif' s'écrit <code>xor</code> le non s'écrit <code>non</code>
Boucle Pour ...de ...jusque ...faire ...Fpour				
Pour I de 1 jusque N faire <i>instructions</i> Fpour	 For(I,1,N) instructions End	 For 1 → I To N instructions Next	Il faudra déclarer auparavant la variable I ➤ Ajouter POUR...DE...A POUR I ALLANT DE 1 A N -DEBUT_POUR -//instructions -FIN_POUR	pour j de 1jusque N faire <i>instructions</i> ; fpour ;  Ne pas utiliser la variable i comme compteur car c'est une lettre prédéfinie qui désigne le i des complexes.

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algbobox	Logiciel Xcas
	Instruction conditionnelle Si...alors...[Sinon]...Fsi			
Si <i>conditions</i> alors <i>instructions</i> Fsi	 CTL If <i>conditions</i> Then <i>instructions</i> End	 COM If <i>conditions</i> Then <i>instructions</i> IfEnd	 Ajouter SI...ALORS SI (condition) ALORS -DEBUT_SI -//instructions -FIN_SI	si <i>conditions</i> alors <i>instructions</i> ; fsi ;
Si <i>conditions</i> alors <i>instructions</i> Sinon <i>instructions</i> Fsi	 CTL If <i>conditions</i> Then <i>instructions</i> Else <i>instructions</i> End	 COM If <i>conditions</i> Then <i>instructions</i> Else <i>instructions</i> IfEnd	 Ajouter SI...ALORS SI (condition) ALORS -DEBUT_SI -//instructions -FIN_SI SINON -DEBUT_SINON -//instructions -FIN_SINON	si <i>conditions</i> alors <i>instructions</i> ; sinon <i>instructions</i> ; fsi ;
Boucle avec arrêt conditionnel Tantque ...faire ...Ftantque				
Tant que <i>conditions</i> faire <i>instructions</i> Ftantque	 CTL While <i>condition</i> <i>instructions</i> End	 COM While <i>condition</i> <i>instructions</i> WhileEnd	 Ajouter TANT QUE... TANT_QUE (condition) FAIRE -DEBUT_TANT_QUE -//instructions -FIN_TANT_QUE	tantque <i>condition</i> faire <i>instructions</i> ; ftantque ;

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algbobox	Logiciel Xcas
	Fonctions mathématiques			
Racine carrée \sqrt{x}	\sqrt{x}	\sqrt{x}	sqrt(x)	sqrt(x)
Puissance x^n	x^n	x^n	pow(x,n)	x^n
Partie entière de x	 NUM int(x)	 NUM Intg(x)	floor(x)	floor(x)
Arrondi à l'unité de x	 NUM round(x,0)	 NUM RndFi RndFix(x,0)	round(x)	round(x)
Reste de la division euclidienne de A par B	$A-B*\text{int}(A/B)$	MOD(A,B) (certaines calculatrices) $A-B*\text{Intg}(A \div B)$	$A\%B$	irem(A,B)
Logarithme népérien de x : $\ln(x)$	ln(x)	ln(x)	log(x)	ln(x)
Exponentielle de e^x	e^x	e^x	exp(x)	exp(x)
Nombre réel pseudo-aléatoire dans $[0; 1[$	 PRB rand	 PROB Rand#	random()	rand(0,1)
Entier aléatoire dans $[a; b]$, avec a et b deux entiers donnés	avec la partie entière : a+int((b-a+1)*rand)	avec la partie entière : a+Intg((b-a+1)*Rand#)	ALGBOBOX_ALEA_ENT(a,b) ou a+floor((b-a+1)*random())	a+rand(b-a+1)