
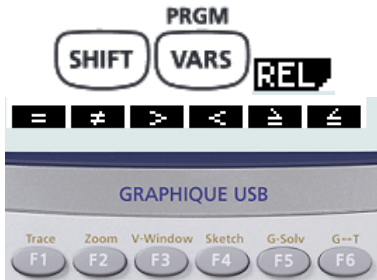

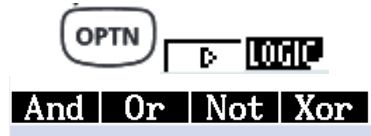


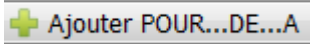



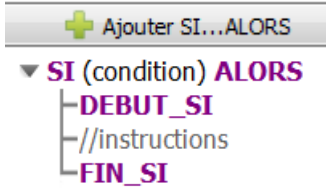


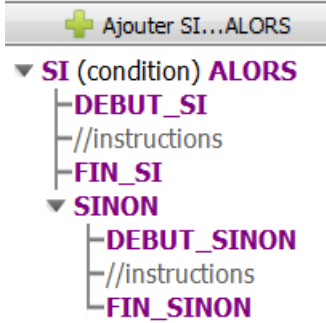


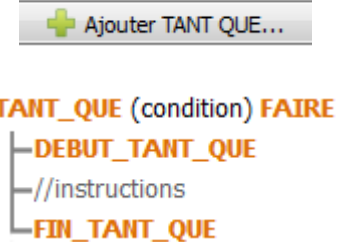
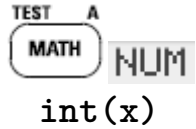
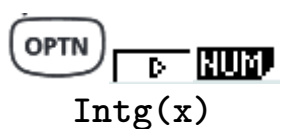
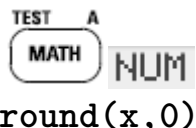
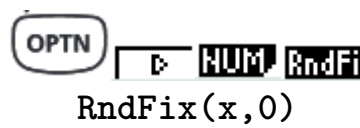
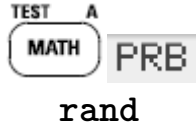





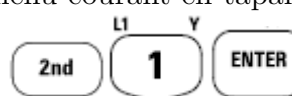


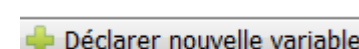
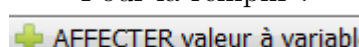

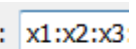


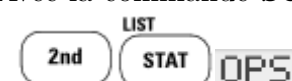

PRINCIPALES COMMANDES POUR PROGRAMMER DANS DIFFÉRENTS LANGAGES

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algobox	Logiciel Xcas
Déclarer une variable A	Inutile	Inutile		<code>local A ;</code>
Saisir A	 Input "A=",A ou Prompt A	 "A=": ? → A		<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 ▴		<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	 $val \rightarrow A$	 $val \rightarrow A$		<code>A:=val;</code>
Utiliser une fonction externe dans un programme	Saisir la fonction dans l'éditeur graphique $Y_1 = X^3 - X - 1$ puis la rappeler dans un programme : $Y_1(\dots)$	Saisir la fonction dans le menu $Y_1 = X^3 - X - 1$ puis la rappeler dans un programme : $Y_1(\dots)$	cliquer sur l'onglet : Saisir la fonction : <input checked="" type="checkbox"/> Utiliser la fonction F1 $F_1(x) = \text{pow}(x,3) - x - 1$ puis la rappeler dans un programme : $F_1(\dots)$	Définir la fonction (3 méthodes) : $f(x) := x^3 - x - 1$ $f := x \rightarrow x^3 - x - 1$ $f := \text{unapply}(x^3 - x - 1, x)$ 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 : $x \rightarrow \dots$

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algobox	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> • "$x \neq 2$" s'écrit <code>x!=2</code> • "$x < 2$" s'écrit <code>x<2</code> • "$x > 2$" s'écrit <code>x>2</code> • "$x \leq 2$" s'écrit <code>x<=2</code> • "$x \geq 2$" s'écrit <code>x>=2</code> 	<ul style="list-style-type: none"> • "$x = 2$" s'écrit <code>x==2</code> • "$x \neq 2$" s'écrit <code>x!=2</code> • "$x < 2$" s'écrit <code>x<2</code> • "$x > 2$" s'écrit <code>x>2</code> • "$x \leq 2$" s'écrit <code>x<=2</code> • "$x \geq 2$" 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) <i>instructions</i> End	 For 1→I To N <i>instructions</i> Next	Il faudra déclarer auparavant la variable I  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 Algobox	Logiciel Xcas
Instruction conditionnelle Si...alors...[Sinon]...Fsi				
Si <i>conditions</i> alors <i>instructions</i> Fsi	 If <i>conditions</i> Then <i>instructions</i> End	 If <i>conditions</i> Then <i>instructions</i> IfEnd		si <i>conditions</i> alors <i>instructions</i> ; fsi ;
Si <i>conditions</i> alors <i>instructions</i> Sinon <i>instructions</i> Fsi	 If <i>conditions</i> Then <i>instructions</i> Else <i>instructions</i> End	 If <i>conditions</i> Then <i>instructions</i> Else <i>instructions</i> IfEnd		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	 While <i>condition</i> <i>instructions</i> End	 While <i>condition</i> <i>instructions</i> WhileEnd		tantque <i>condition</i> faire <i>instructions</i> ; ftantque ;

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

Langage algorithmique	Langages de programmation			
	Sur TI	Sur Casio	Logiciel Algobox	Logiciel Xcas
Listes				
Créer et remplir une liste	<p>Les listes L_1, L_2, \dots, L_2 existent déjà dans le mode</p> <p>STAT :  </p> <p>On peut donc les remplir directement avec ce menu.</p> <p>Cela peut se faire aussi dans le menu courant avec la commande :</p> $\{x_1, \dots, x_n\} \rightarrow L_1$ <p>On peut l'afficher dans le menu courant en tapant :</p> 	<p>Les listes List 1, List 2, ..., List 26 existent déjà dans le menu STAT :</p>  <p>On peut donc les remplir directement avec ce menu.</p> <p>Cela peut se faire aussi dans le menu courant avec la commande :</p> $\{x_1, \dots, x_n\} \rightarrow \text{List 1}$ <p>On peut l'afficher dans le menu courant en tapant :</p> 	<p> puis préciser le type Liste.</p> <p>Pour la remplir :</p> <p></p> <p>puis</p> <p> prend la valeur : </p> <p>le de la liste : </p> <p>en mettant 1 au rang de la liste et en séparant chaque valeur par :</p> <p>Pour afficher le contenu d'une liste, on utilise une boucle.</p>	<p>Pour créer une liste $L := [x_1, \dots, x_n]$</p> <p>Pour afficher le contenu d'une liste :</p> <p>retourne L</p>
Élément de rang k d'une liste	<p>Le premier rang d'une liste L_1 est 1 et le dernier rang est Dim(L₁).</p> <p>$L_1(k)$ est le terme de rang k de la liste 1.</p>	<p>Le premier rang d'une liste List 1 est 1 et le dernier rang est Dim List 1.</p> <p>List 1[k] est le terme de rang k de la liste 1.</p>	<p>L[1] est le premier terme de la liste L (on peut débiter à 0 : L[0]).</p> <p>L[k] est le terme de rang k de la liste L.</p> <p>La longueur d'une liste commençant à 1 est donnée par L.length-1</p>	<p> L[0] ou L(1) désignent le premier terme de la liste L.</p> <p>L[k] est le terme de rang k de la liste L donc le $(k + 1)$-ème terme de cette liste</p> <p>La longueur d'une liste est donnée par dim(L)</p>
Remplir une liste avec p entiers aléatoires pris dans $[a; b]$, avec a et b deux entiers donnés	<p>Avec la commande seq</p>  <p>seq(a+int((b-a+1)*rand),K,1,p,1) → L₁</p>	<p>Avec la commande Seq</p>  <p>seq(a+Intg((b-a+1)*Rand#),K,1,p,1) → List 1</p>	<p>Il faut créer une boucle pour remplir la liste terme après terme :</p> <p>POUR k ALLANT_DE 1 A p</p> <p> DEBUT_POUR</p> <p> L[k] PREND_LA_VALEUR ALGOBOX_ALEA_ENT(a,b)</p> <p> FIN_POUR</p>	<p>L :=</p> <p>[(a+rand(b-a+1))\$(k=1..p)]</p>