

Algorithmique

Définition : un algorithme est un ensemble d'instructions destiné à résoudre un problème donné de façon systématique. Les instructions peuvent être effectuées par une calculatrice ou un ordinateur, à condition de connaître la syntaxe permettant de programmer. Une fois l'algorithme implémenté, tout utilisateur de la machine peut résoudre le problème donné au départ.

Exemple : le problème est : "quel est le carré de ce nombre?" Cet algorithme est déjà programmé dans votre calculatrice et est effectué à l'appui sur la touche x^2 . L'utilisateur tape par exemple le nombre 3, puis appuie sur la touche x^2 pour exécuter le programme, la solution est alors donnée :

I. Structure générale

Un algorithme contient, en général, 3 étapes successives:

Début de l'algorithme

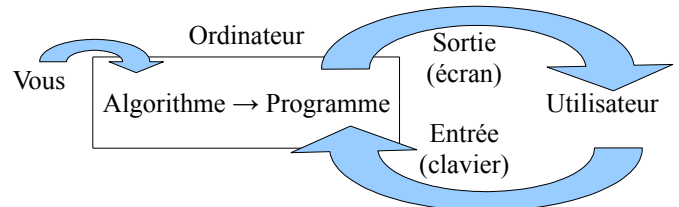
1) entrée des valeurs utilisées dans le problème posé

2) calculs, instructions permettant de résoudre le

problème

3) sortie des valeurs solutions du problème

Fin de l'algorithme



Casio	TI
Menu, PRGM, NEW, entrer un nom de programme.	PRGM sélectionner NEW puis entrer un nom de programme.
Les mots sont obtenus par : SHIFT PRGM puis en sélectionnant par les touches F1 etc. L'affichage signifie que d'autres choix sont disponibles.	Les mots sont obtenus avec PRGM , en se déplaçant dans les menus à l'aide des touches directionnelles, puis en sélectionnant à l'aide de ENTER .
Les caractères sont obtenus par ALPHA puis la touche désirée	Les caractères sont obtenus par ALPHA puis la touche désirée

Les valeurs utilisées sont stockées dans des zones mémoires des machines appelées variables. Le programmeur doit attribuer un nom à chaque variable utilisée dans le programme pour pouvoir utiliser ces noms dans les instructions.

Langage naturel	Algobox	Python	Casio	TI
entrer la valeur de a	AFFICHER "Donner a" LIRE a	a=input("Donner a")	CHAR EYBL "Donner a":? A	PRGM I/O Disp "Donner a" Prompt A
affecter à b la valeur ...	b PREND_LA_VALEUR...	b=...	... B	... STO B
sortir la valeur de b	AFFICHER b	Print "b=",b	"b=":B ↵	Disp "b=",B

Remarque : certains langages de programmation (Algobox par exemple) nécessitent une déclaration préalable des noms des variables utilisés dans le programme.

II. Instructions conditionnelles

Langage naturel	Algobox	Python	Casio	TI
Si <i>condition</i> alors <i>instructions1</i> sinon <i>instructions2</i>	▼ SI (condition) ALORS DEBUT_SI //instructions1 FIN_SI ▼ SINON DEBUT_SINON //instructions2 FIN_SINON	if <i>condition</i> : <i>instructions1</i> then : <i>instructions2</i> un bloc d'instructions a la même marge	SHIFT PRGM COM If <i>condition</i> Then <i>instructions1</i> Else <i>instructions2</i> IfEnd	PRGM CTL If <i>condition</i> Then <i>instructions1</i> Else <i>instructions2</i> End
Tests : = , ≠ , ≥ , ≤	= , ≠ , >= , <=	= , ≠ , >= , <=	SHIFT PRGM COM REL	2nd MATH

III. Boucles

Langage naturel	Algobox	Python	Casio	TI
Effectuer <i>n</i> fois , les <i>instructions</i>	▼ POUR i ALLANT_DE 1 A n DEBUT_POUR //instructions FIN_POUR	for i in range(1,n+1): <i>instructions</i>	SHIFT PRGM COM ↵ For 1→I To n <i>instructions</i> Next	PRGM CTL For (I,1,n) <i>instructions</i> End
Tant que <i>condition</i> Effectuer les <i>instructions</i>	▼ TANT_QUE (condition) FAIRE DEBUT_TANT_QUE //instructions FIN_TANT_QUE	While <i>condition</i> : <i>instructions</i>	SHIFT PRGM COM ↵ ↵ While <i>condition</i> <i>instructions</i> WhileEnd	PRGM CTL While <i>condition</i> <i>instructions</i> End

Commandes accessibles dans PRGM sur Casio

COM	If Then Else End For To Step Next While Wend Do Lp-While
CTL	Prog Rtn Brk Stop
JUMP	Lbl Goto $\Rightarrow$ Isz Dsz
?	
▲	
CLR	Text Grph List Mat
DISP	Stat Grph Dyna F-Tbl R-Tbl
REL	= $\neq$ > < $\geq$ $\leq$
I/O	Lcte Gtkey Send Recv S3B R3B Open Close
:	

Caractères spéciaux : **CHAR** puis **MATH** **SYBL** **ABF** **αβγ**

Appel d'une fonction : **VAR** **GRPH** **Y** puis touche du numéro de la fonction

Commandes accessibles dans PRGM sur TI

CTL	I/O	EXEC
1 : If 2 : Then 3 : Else 4 : For(5 : While 6 : Repeat 7 : End 8 : Pause 9 : Lbl 0 : Goto A : ISX(B : DSX(C : Menu(D : prgm E : Return F : Stop G : DelVar H : GraphStyle(	1 : Input 2 : Prompt 3 : Disp 4 : DispGraph 5 : DispTable 6 : Output(7 : getKey 8 : ClrHome 9 : ClrTable 0 : GetCalc(A : Get(B : Send(	Liste des programmes

Commandes accessibles dans TEST sur TI : **2nd** puis **TEST** **MATH**

TEST	LOGIC
1 : = 2 : $\neq$ 3 : > 4 : $\geq$ 5 : < 6 : $\leq$	1 : and 2 : or 3 : xor 4 : not(

Caractère " : **ALPHA** puis **MEM** **+**

Appel d'une fonction : **VAR** **Y-VARS** **1:Function...** **1:Y1**