

Algorithmique

Programmes relatifs à l'algorithme vu en exemple :

Casio	Texas	AlgoBox	Xcas
<pre>=====SI ALORS= ?+N# If N<75# Then N*0.16# Else N*0.12# IfEnd If Then Else IfEnd</pre>	<pre>PROGRAM:SI :Input N :If N<75 :Then :Disp N*0.16 :Else :Disp N*0.12 :End</pre>	<pre>VARIABLES - N EST DU TYPE NOMBRE DEBUT ALGORITHME - LIRE N - SI (N<75) ALORS - DEBUT SI - N PREND LA VALEUR N*0.16 - AFFICHER N - FIN SI - SINON - DEBUT SINON - N PREND LA VALEUR N*0.12 - AFFICHER N - FIN SINON</pre>	<pre>saisir(N); Si N < 75 alors afficher(N*0.16); sinon afficher (N*0.12); fsi</pre>

Si le nombre de photocopies est 25, la dépense est 4 €, et s'il est de 100, la dépense est 12 €.

Programmation d'un calcul itératif avec fin de boucle conditionnelle

Exemple : Une balle lâchée d'une hauteur donnée rebondit chaque fois qu'elle touche le sol aux $\frac{1}{5}$ ^e de sa hauteur.
Écrire un algorithme qui donne le nombre de rebonds de la balle avant que celle-ci ne soit à un millimètre du sol.

On appelle X la variable donnant la hauteur en millimètres atteinte par la balle après chaque rebond. On doit ici répéter l'instruction « X prend la valeur de $\frac{X}{5}$ », mais on ne connaît pas à l'avance le nombre de répétitions. On teste alors une condition en début de boucle ($X > 1$ ici) et le traitement dans la boucle n'est réalisé que si la condition est vérifiée.
On introduit un « compteur » R pour compter le nombre de rebonds : on l'initialise à 0, et chaque fois que la boucle est parcourue, ce compteur est augmenté d'une unité.

Variables	
X , hauteur de la balle	On entre dans la variable X , la hauteur initiale de la balle.
Initialisation	
Saisir X	On initialise à 0 le nombre R de rebonds de la balle.
Traitement	
R prend la valeur 0	
Tant que $X > 1$	On entre dans la boucle si X dépasse 1 mm.
X prend la valeur $X/5$	La hauteur est divisée par 5 après un rebond.
R prend la valeur $R + 1$	Il y a un rebond de plus, donc R augmente d'une unité.
Fin Tant que	On revient au début de la boucle pour tester si la hauteur de la balle dépasse ou non 1 mm.
Sortie	
Afficher R	On affiche le nombre R de rebonds.

Avec une hauteur initiale de 2 mètres, le fonctionnement de l'algorithme peut être représenté à l'aide d'un tableau. Les valeurs des variables X et R pour les différentes étapes sont données ci-contre :

Tant que {condition C}
Faire {instructions}
Fin Tant que

Étapes	X	R
0	2 000	0
1	400	1
2	80	2
...

C'est la **structure itérative avec fin de boucle conditionnelle** qui permet de résoudre ce problème.

Remarque : Dans l'instruction « Tant que ... », la condition C est testée en début de boucle, donc si la condition n'est pas vérifiée au départ, la boucle n'est jamais exécutée.

Programmes relatifs à l'algorithme vu en exemple :

Casio	Texas	AlgoBox	Xcas
<pre>=====TANT QUE===== ?→X A←R While X>1 X←X/5 R←R+1 WhileEnd R</pre>	<pre>PROGRAM:TANT :Input X :0←R :While X>1 : X←X/5 : R←R+1 :End :Disp R</pre>	<pre>VARIABLES ├── X EST DU TYPE NOMBRE ├── R EST DU TYPE NOMBRE └── DEBUT_ALGORITHME ├── LIRE X ├── R PREND LA VALEUR 0 └── TANT_QUE (X>1) FAIRE ├── DEBUT_TANT_QUE ├── X PREND LA VALEUR X/5 ├── R PREND LA VALEUR R+1 ├── FIN_TANT_QUE └── AFFICHER R</pre>	<pre>saisir(X); R:=0; tantque X>1 faire X:=X/5; R:=R+1; ftantque; afficher(R)</pre>

Si la balle est lâchée d'une hauteur de 2 m, soit 2 000 mm, la balle effectuera 5 rebonds avant d'être à 1 mm du sol.

Récapitulatif des instructions utilisées en programmation

	Casio	Texas	AlgoBox	Xcas
Créer un nouveau programme	Menu PRGM puis NEW (touche F3)	prgm puis NOUV	Cliquer sur Nouveau	Menu File puis Nouveau programme
Entrer A Afficher « A = ? »	? → A « A = » ? → A	Input A Prompt A	+ AJOUTER Lire Variable	saisir(A)
Afficher A	A	Disp A	+ AJOUTER Afficher Variable	afficher(A)
Afficher un texte, par exemple « oui »	« oui »	Disp « oui »	+ AJOUTER Message	afficher « oui »
Opérateurs relationnels (comme =, ≤, ≥, ≠)	PRGM (touches SHIFT VARS), puis ▷ , puis REL	TEST (touches 2nde math)	Au clavier	Au clavier
Affecter à B la valeur de A	Taper A puis la touche → et enfin B	Taper A puis la touche STO→ et enfin B	+ AFFECTER valeur à variable	B:=A
Pour I variant de I ₀ à N Faire {instructions} Fin Pour	For I ₀ → I to N {instructions} Next	For (I, I ₀ , N) {instructions} End	+ AJOUTER POUR... DE...A	pour I de I ₀ jusque N faire {instructions}; fpour
Si {condition C} Alors {instructions A} Sinon {instructions B} FinSi	If {condition C} Then {instructions A} Else {instructions B} IfEnd	If{condition C} Then{instructions A} Else {instructions B} End	+ AJOUTER SI...ALORS	si {condition C} alors {instructions A} sinon {instructions B} fsi
Tant que {condition} Faire {instructions} Fin Tant que	While {condition C} {instructions} WhileEnd	While {condition C} {instructions} End	+ AJOUTER TANT QUE...	tant que {condition C} Faire {instructions} ftantque