

CONTENU DU DOSSIER

1. Découvrir l'algorithmique : petits algorithmes à tester pas à pas.
 - Affecter des variables
 - Le programme SOMME
2. Un test logique :
 - SI ... ALORS : programme DELTA
3. Les boucles :
 - La boucle POUR : programme SUITE arithmétique
 - La boucle TANT QUE : programme BASE b
4. Un programme sur tableur : l'algorithme d'Euclide.
5. Sensibiliser à la vitesse d'exécution :
 - Nombre d'opérations effectuées dans un calcul : programme HORNER (calcul de l'image d'un réel par un polynôme)
 - Résolution d'une équation par DICHOTOMIE (à comparer à la méthode par balayage)
6. Réinvestir ces connaissances : la suite de SYRACUSE.

AFFECTER UNE VARIABLE

Qu'obtient-on à l'écran lorsque l'on programme l'algorithme suivant ? (Tester l'algorithme pas à pas)

Demander A
Demander B
Affecter la valeur de A à B
Affecter la valeur de B à A
Afficher A
Afficher B

EFFECTUER UNE SOMME

Qu'obtient-on à l'écran lorsque l'on programme les algorithmes suivants ? (Tester l'algorithme pas à pas)

exemple 1 :

Entrée	Demander A
Traitement	Affecter la valeur de A à S Faire 10 fois la procédure suivante Affecter la valeur de S+1 à S
Sortie	Afficher S

exemple 2 :

Donner à S la valeur 0

Début de la boucle

Demander A

Donner à S la valeur S+A

Recommencer au début de la boucle jusqu'à ce que A=F

Afficher S

La boucle SI ... ALORS

Pour un polynôme donné $P(x) = ax^2 + bx + c$, le programme suivant affiche le discriminant D et les solutions éventuelles. S'il n'y en a pas, il affiche "Pas de solutions".

Compléter l'algorithme :

Demander A
 Demander ...
 Demander ...
 Affecter la valeur de à D
 Afficher ...

Si $D > 0$
Alors affiche
Sinon Si $D = 0$
 Alors
 Sinon

Voici cet algorithme traduit en langage machine :

Commentaires	TI	CASIO
<p>Il faut avant tout ouvrir une page pour écrire le programme. La calculatrice attend que vous donniez un nom au programme. Le curseur est un A, ce qui veut dire qu'elle fonctionne en mode alphabétique, comme une machine à écrire. Vous pouvez maintenant commencer l'écriture du programme.</p> <p>La calculatrice vous demandera de rentrer les valeurs A, B et C et les gardera en mémoire. Elle calcule Delta et le garde en mémoire sous le nom de D. Elle affiche Delta. Si $\Delta > 0$</p> <p>Alors affiche qu'il y a deux solutions. Elle calcule ces deux solutions puis les affiche sous forme de fraction.</p> <p>Sinon Si $\Delta = 0$ Alors La calculatrice affiche qu'il y a une solution, Elle affiche cette solution Sinon Elle affiche qu'il n'y a pas de solution. Fin de la boucle Si...alors.</p>	<pre> PRGM DELTA NEW Ecrire : DELTA Puis taper sur ENTER La calculatrice affiche : PROGRAM :DELTA :PROMPT A,B,C :B²-4AC → D :DISP "DELTA",D :If D>0 :Then :Disp "2 sol" :Disp (-B-√D)/(2A) ▶Frac, (-B+√D)/(2A)▶Frac :Else :If D=0 :Then :Disp "1 sol", Disp -B/(2A)▶Frac :Else :Disp "0 sol" :End </pre>	<pre> MENU PRGM NEW Ecrire : DELTA Puis taper sur EXE La calculatrice affiche : ==== DELTA ==== "A?":? → A↓ "B?":? → B↓ "C?":? → C↓ B²-4AC → D↓ "D=":D▲ If D>0↓ Then "2 sol"↓ "X=": (-B-√D)÷(2A)▲ "Y=": (-B+√D)÷(2A)↓ Else If D=0↓ Then "1 sol"↓ "X=": -B÷(2A)↓ Else "0 sol"↓ I-End↓ I-End </pre>
<p>Quitter après le dernier mot du programme sans rien écrire de plus.</p>	<pre> QUIT </pre>	<pre> EXIT </pre>

Pour exécuter le programme.

TI	CASIO
<p>PRGM Choisir le programme DELTA puis ENTER La calculatrice affiche prgmDELTA Faire ENTER La calculatrice affiche A? Entrer la valeur de A, ENTER De même avec B et C. A la fin la calculatrice affiche Done</p>	<p>MENU PRGM Choisir le programme DELTA puis EXE La calculatrice affiche A? Entrer la valeur de A, EXE De même avec B et C. A la fin la calculatrice affiche FIN</p>

Où trouver les commandes pour écrire les programmes ?

TI	CASIO
Commandes pour les caractères alphabétiques	
<p>Les lettres et les guillemets " " sont sur le clavier de la calculatrice et s'obtiennent avec la touche ALPHA comme tous les symboles en haut à droite des touches. A-LOCK bloque la calculatrice en mode alphabétique. Attention ! Les « espaces » que vous voyez dans ce programme ne sont pas obtenus par la la touche ALPHA □ de votre clavier mais sont naturellement associés aux instructions des touches de programmes.</p>	<p>Les lettres sont sur le clavier de la calculatrice et s'obtiennent avec la touche ALPHA comme tous les symboles en haut à droite des touches. A-LOCK bloque la calculatrice en mode alphabétique. Les guillemets " " sont obtenus en appuyant sur ALPHA mais au bas de l'écran. Attention ! Les « espaces » que vous voyez dans ce programme ne sont pas obtenus par la touche ALPHA □ de votre clavier mais sont naturellement associés aux instructions des touches de programmes.</p>
Commandes pour les caractères mathématiques	
<p>« / » est la division ÷ « * » est la multiplication × « - » est le signe opératoire de la soustraction. A ne pas confondre avec « (-) » qui se met en début de calcul et désigne le signe du premier nombre. « > » est dans le menu TEST. ►Frac est dans le menu MATH.</p>	<p>« - » est le signe opératoire de la soustraction. A ne pas confondre avec « (-) » qui se met en début de calcul et désigne le signe du premier nombre. « > » est dans le menu PRGM ▷ REL. En règle générale, penser à appuyer sur la ▷ en bas à droite de l'écran si vous ne trouvez pas une commande.</p>
Commandes pour les instructions usuelles des programmes	
<p>« : » s'inscrit dans un programme à chaque fois que vous taper sur ENTER pour aller à la ligne. Prompt qui permet d'entrer les valeurs et de les mettre en mémoire se trouve dans le menu PRGM puis I/O. Ou bien DISP suivi de INPUT. → s'obtient en appuyant sur STO, en bas à gauche de votre clavier de calculatrice.</p>	<p>« ↵ » s'inscrit dans un programme à chaque fois que vous taper sur EXE pour aller à la ligne. ? qui signifie « invite moi à entrer une valeur » (et qui sur certaine calculatrice se traduit par un ? lorsque le programme est exécuté) se trouve dans le menu PRGM (obtenu par SHIFT VARS) puis ▷. ▲ est un ordre que vous donnez à la calculatrice et signifie : « affiche à l'écran la valeur calculée » (lorsqu'il s'agit d'un calcul). Il se trouve lui aussi dans le menu PRGM puis ▷. → est sur le clavier.</p>
Commandes particulières pour les « boucles » et les « tests »	
<ul style="list-style-type: none"> • IF (en français SI), THEN (en français ALORS) , ELSE (en français SINON) • WHILE (TANT QUE) <p>se trouvent dans le menu PRGM puis CTL (il s'agit des contrôles).</p>	<ul style="list-style-type: none"> • IF (en français SI), THEN (en français ALORS) , ELSE (en français SINON), I-End (fin de la boucle SI) • WHILE (TANT QUE) <p>se trouvent dans le menu 2nd PRGM puis COM (il s'agit des commandes).</p>
<p>Pour revenir à un menu principal, penser à utiliser la touche 2nd QUIT.</p>	<p>Pour revenir à un menu principal, penser à utiliser la touche EXIT.</p>

Pour effacer un programme.

TI	CASIO
<p>$\boxed{2^{\text{nd}}}$ $\boxed{\text{mem}}$ puis choix 2 : mem Mgmt/Del et choix 7 : prgm Placer le curseur devant le ou les programme(s) à effacer et appuyer sur $\boxed{\text{enter}}$ puis $\boxed{\text{DEL}}$ et YES.</p>	<p>Dans le menu PROG, placer le curseur sur le programme à effacer et choisir le menu DEL.</p>

SUITES

Le programme suivant calcule les termes successifs d'une suite arithmétique, lorsqu'on entre le premier terme u_0 , la raison et le nombre de termes souhaités.

Algorithme :

Demander u_0
 la raison r
 le rang n du dernier terme

Pour i allant de 1 à n
 Affecter la valeur de $U + r$ à U
 Afficher U

Voici comment cet algorithme peut se traduire en langage machine :

<u>Commentaires</u>	TI	CASIO
TI : Pause permet d'arrêter le défilement des valeurs	Disp "U0" Input U Prompt R Prompt N For(I,1,N) U+R → U Disp U Pause End	"U0":? → U "R":? → R "N":? → N For 1 → I To N U+R → U U▲ Next

La boucle TANT QUE

Le programme suivant donne l'écriture en base B d'un nombre entier N écrit en base 10.

1. Déterminer l'écriture de 2123 en base 5.
2. Donner les valeurs successives du dividende N, du quotient Q et du reste R.

Etapes	N (dividende)	Q (quotient)	R (reste)
$2123 = 424 \times 5 + 3$			
$424 = 84 \times 5 + 4$			
$84 = 16 \times 5 + 4$			
$16 = 3 \times 5 + 1$			
$3 = 0 \times 5 + 3$			

3. Qu'est ce qui provoque l'arrêt de cette suite de divisions ? Autrement dit, quel est le *test d'arrêt* ?
4. La fonction **INT** de la calculatrice donne la partie entière d'un nombre.

A l'aide de cette fonction, comment obtient-on le quotient de la division euclidienne de N par B ?

En déduire le reste de la division.

5. Compléter l'algorithme :

Demander ...

Demander ...

Tant que **faire**

Affecter la valeur de à Q

Affecter la valeur de à R

Afficher

Affecter la valeur de à N

Fin TantQue

Voici comment cet algorithme peut se traduire en langage machine :

Commentaires	TI	CASIO
<p>Il faut avant tout ouvrir une page pour écrire le programme. La calculatrice attend que vous donniez un nom au programme. Le curseur est un A, ce qui veut dire qu'elle fonctionne en mode alphabétique, comme une machine à écrire. Vous pouvez maintenant commencer l'écriture du programme.</p> <p>La calculatrice vous demandera de rentrer une valeur N puis une valeur B. Elle stockera ces valeurs dans la mémoire N et dans la mémoire B.</p> <p>Tant que le contenu de Q est strictement positif, la calculatrice remplace le nombre N par Q. Elle fait la division euclidienne de N par B et ne garde que la partie entière. Elle calcule le reste de la division euclidienne de N par B et le stocke dans la mémoire R. La calculatrice arrête le programme lorsque $Q=0$.</p>	<pre> PRGM NEW Ecrire : BASE Puis taper sur ENTER La calculatrice affiche : PROGRAM :BASE :DISP "N" :INPUT N :DISP "B" :INPUT B :N → Q :While Q>0 : Int (N/B) → Q : N-Q*B → R : Disp R : Pause : Q → N :End </pre>	<pre> MENU PRGM NEW Ecrire : BASE Puis taper sur EXE La calculatrice affiche : ====BASE ==== "N" ↓ ? → N ↓ "B" ↓ ? → B ↓ N → Q While Q>0 ↓ Int (N/B) → Q ↓ N-Q×B → R ↓ R ↓ Q → N ↓ WhileEnd ↓ "FIN" ↓ Stop </pre>
<p>Quitter après le dernier mot du programme sans rien écrire de plus.</p>	<pre> QUIT </pre>	<pre> QUIT </pre>

TESTER UN ALGORITHME

Voici un algorithme de passage de la base 10 à la base B :

```
DEBUT
Nombre → N
Base → B
0 → I
0 → A
TANT QUE N > 0
    A + RESTE(N/B) × 10I → A
    QUOTIENT de la division de N par B → N
    I + 1 → I
FIN TANT QUE
AFFICHER (A)
FIN
```

Questions :

1. Tester cet algorithme pour $N=111$ et $B=5$ (Ecrire toutes les étapes).
2. Précisez ce que l'utilisateur obtient sur l'écran de la calculatrice pour un nombre N .
3. Cet algorithme fonctionne-t-il pour une base supérieure ou égale à 10 ? Justifier.

PROGRAMMATION SUR TABLEUR

Indication : Vous pourrez utiliser la fonction MOD : MOD(nombre ; diviseur) renvoie le reste d'une division.

Voici une feuille de calcul sur tableur :

	A	B	C
1	a	b	reste
2	4998	4746	252
3	4746	252	210
4	252	210	42
5	210	42	0
6	42	0	#DIV/0!
7	0	#DIV/0!	#DIV/0!
8	#DIV/0!	#DIV/0!	#DIV/0!

1. Que permet-elle de calculer ?
2. Quelle formule a été écrite en A3, B3 et C2 ?
3. Comment obtient-on les résultats dans les plages de cellules (A4 : A8) et (C3 : C8) ?

Expliquez le message d'erreur qui apparaît dans cette feuille.

Hörner

Remarque : Soit $P(x)$ un polynôme de degré $n = 3$. $P(x)$ peut s'écrire sous la forme suivante :

$$\begin{aligned}
 P(x) &= a_0 + a_1x + a_2x^2 + a_3x^3 \\
 &= a_0 + x(a_1 + a_2x + a_3x^2) \\
 &= a_0 + x(a_1 + x(a_2 + xa_3))
 \end{aligned}$$

On peut utiliser cette forme pour calculer l'image d'un réel x_0 par P .
C'est la méthode de Hörner.

Algorithme :

Demander le degré n du polynôme
 la valeur de x_0
 les coefficients du polynômes

Affecter la valeur de a_n à A

|| Pour i allant de n à 1
 || affecter la valeur de $Ax + a_{i-1}$ à A

Afficher A.

Voici comment cet algorithme peut se traduire en langage machine :

Commentaires	TI	CASIO
<p>TI : Vide la liste L_1 du mode statistique.</p> <p>casio : Annonce la dimension de la liste. Demande les coefficients du polynôme puis les met dans la liste 1. TI : L'utilisateur devra entrer les coefficients sous cette forme : $\{2;1;3\}$ casio : donner les coefficients dans l'ordre $a_0; a_1; a_2 \dots$</p> <p><i>remarque :</i> les coefficients sont indicés de 0 à n mais les éléments de la liste sont numérotés de 1 à $n+1$.</p> <p>Lorsque I décroît, il faut préciser le pas : -1</p>	<p>prompt N,X</p> <p>clrlist L₁</p> <p>Disp "$\{a_0; a_1; a_2; \dots\}$"</p> <p>input L₁</p> <p> </p> <p>$L_1(N+1) \rightarrow A$</p> <p> for (I, N+1, 2, -1) $A \times X + L_1(I-1) \rightarrow A$ end</p> <p>Disp A</p>	<p>"N" : ? \rightarrow N "X" : ? \rightarrow X</p> <p> </p> <p>N+1 \rightarrow Dim List1</p> <p> for 1 \rightarrow I to N+1 "coef ?" : ? \rightarrow List1[I] next</p> <p> </p> <p>List1[N+1] \rightarrow A</p> <p> for N+1 \rightarrow I to 2 step -1 $A \times X + List1[I-1] \rightarrow A$ Next</p> <p> </p> <p>A</p>

Questions :

1. Tester cet algorithme avec $P(x) = 2 - 4x + 3x^2 + 2x^3 - 0,5x^4$ et $x_0 = 7$.
2. Ecrire ce polynôme sous la forme présentée en introduction puis déterminer le nombre d'opérations effectuées (additions et multiplications) pour calculer $P(7)$.
3. Compter le nombre d'opérations effectuées pour le calcul suivant :

$$P(7) = 2 - 4 \times 7 + 3 \times 7^2 + 2 \times 7^3 - 0,5 \times 7^4$$

4. Ecrire l'algorithme qui permet de calculer l'image de x_0 comme dans le calcul précédent.

TI	CASIO
prompt N,X	"N" : ? → N "X" : ? → X
clrlist L ₁	
Disp "{a ₀ ; a ₁ ; a ₂ ; ...}"	N+1 → Dim List1
input L ₁	for 1 → I to N+1 "coef ?" : ? → List[1] next
L ₁ (1) → A	List1[1] → A
for (I, 2,N+1) L ₁ (I) × X ^{I-1} + A → A end	for 2 → I to N+1 List1[I] × X ^{I-1} + A → A Next
Disp A	A

Pour information :

Pour un polynôme de degré n :

- Le nombre d'opérations avec l'écriture $a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$ est de l'ordre de

$$\frac{n^2}{2} \left(\frac{n^2}{2} + \frac{3}{2}n \right)$$

- Le nombre d'opérations avec l'écriture $a_0 + x(a_1 + x(a_2 + x(a_3 + \dots (a_{n-1} + xa_n) \dots))$ est de l'ordre de $2n$.

Dichotomie

Le programme suivant donne un encadrement d'amplitude choisie d'une solution de l'équation $f(x)=0$.

Il nécessite de connaître un intervalle où se situe la solution.

Algorithme :

Demander l'amplitude E
les bornes de l'intervalles : A et B
la fonction f .

Tant que l'amplitude de l'intervalle [A;B] est supérieure à E :
Déterminer le centre de l'intervalle [A;B]
Affecter cette valeur à C.
Regarder si la solution se situe dans l'intervalle [A ; C] ou [C ; B]
Recommencer avec l'intervalle où se situe la solution.

Afficher A et B.

Voici comment cet algorithme peut se traduire en langage machine :

Commentaires	TI	CASIO
<p>TI : L'utilisateur entrera la fonction avec des guillemets. Par exemple : " 2X-3 "</p> <p>casio : L'utilisateur devra aller dans le mode GRAPH entrer la fonction avant d'executer le programme.</p> <p>casio : Pour calculer l'image de C par Y1 : C → X Y1 → Y</p>	<pre>input "Y1=", Y1 prompt E, A, B While abs(A-B)>E (A+B)/2 → C if Y1(C)×Y1(A) ≤ 0 then C → B else C → A end End Disp A, B</pre>	<pre>"A":? → A "B":? → B "E":? → E While Abs(A-B)>E:Do (A+B)/2 → C C → X Y1 → Y A → X Y1 → Z If Y×Z ≤ 0 Then C → B Else C → A IfEnd WhileEnd A B Stop</pre>

Questions :

1. Tester cet algorithme pas à pas avec $E = 0,1$; $I = [0;2]$; $f(x) = 2x - 3$.

Quelles sont les valeurs de A et B affichées ?

2. $A = 1,4375$ et $B = 1,5$.

- a) Donner une valeur approchée par excès à 10^{-1} de la solution.
- b) Donner une valeur approchée par défaut à 10^{-1} de la solution.
- c) Peut-on donner une valeur arrondie à 10^{-1} de la solution avec ces valeurs de A et B ? Justifier.
- d) Quelle amplitude faut-il prendre pour avoir une valeur arrondie à 10^{-1} de la solution ?

3. La solution de l'équation $f(x) = 0$ est $\frac{3}{2}$. Quelle instruction peut-on ajouter à la fin du programme pour qu'il ne donne que A (ou que B) lorsque c'est la solution exacte ?

Suite de Syracuse

La suite de Syracuse est définie par $u_0 \in \mathbb{N}^*$ et

$$\text{Pour tout } n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

- 1) Calculer les premiers termes de la suite pour $u_0 = 1$; $u_0 = 3$ et $u_0 = 7$.

On ne sait pas à l'heure actuelle s'il existe un entier u_0 pour lequel cette suite n'atteint jamais 1.

- 2) Ecrire un programme demandant u_0 et n à l'utilisateur et affichant toutes les valeurs $u_1; u_2, \dots, u_n$.
- 3) Ecrire un programme demandant u_0 à l'utilisateur et affichant toutes les valeurs $u_1; u_2, \dots, u_N$ où N est le plus petit entier $k \in \mathbb{N}^*$ tel que $u_k = 1$.
- 4) Modifier votre programme pour qu'il affiche en plus N et le plus grand élément de la séquence (u_0, \dots, u_N)

Réponses :

- 1) Pour $u_0 = 1$: $u_1 = 4, u_2 = 2, u_3 = 1$ $u_0=1, u_1=4, u_2=2, u_3=1 \dots$

- 2) Demander u_0
Demander n
Affecter la valeur de u_0 à U

Pour i allant de 1 à N

Si U est pair

Affecter la valeur de U/2 à U

Sinon

Affecter la valeur de 3U+1 à U

Fin Si

Afficher U

Fin de la boucle

Afficher U

- 3) Demander u_0
Affecter la valeur de u_0 à U

Tant que $U \neq 1$

Si U est pair

Affecter la valeur de $U/2$ à U

Sinon

Affecter la valeur de $3U+1$ à U

Fin Si

Afficher U

Fin Tant que

- 4) Demander u_0
Affecter la valeur de u_0 à U
Affecter la valeur 0 à N
Affecter la valeur de U à V

Tant que $U \neq 1$

Affecter la valeur de $N+1$ à N

Si U est pair

Affecter la valeur de $U/2$ à U

Sinon

Affecter la valeur de $3U+1$ à U

Fin Si

Afficher U

Si $U > V$

Affecter la valeur de U à V

Fin Tant que

Afficher V

Afficher N