2de TD n°3 : utilisation de la *calculatrice* en mode « Programme »

Algorithmes et programmes : Un *algorithme* est un ensemble d'instructions structuré de manière à atteindre un but. Ces instructions manipulent des données (nombres, textes, etc.) : entrées, stockages en mémoire, calculs, affichages. Comme un algorithme est souvent exécuté par une machine (calculatrice, ordinateur, robot), il doit être traduit dans un langage que cette machine comprend : les *programmes* pour calculatrices utilisent une syntaxe dérivée du langage « basic » (le basic Casio est différent du basic TI) tandis que les programmes en « python » (un autre langage de programmation) s'exécutent sur ordinateur. Beaucoup d'autres langages de programmation existent (java, C, ...).

Le mode « programmation » de la calculatrice :

Casio : On y entre avec la touche PRGM. Pour commencer un nouveau programme, sélectionner NEW (bouton F3), taper ensuite un nom pour le programme (ALGO1 par exemple) et EXE. Il ne reste plus qu'à écrire le programme. Pour modifier un programme existant, choisir EDIT dans le menu PRGM ; pour l'exécution, choisir EXE.

TI : On y entre avec la touche « prgm ». Pour commencer un nouveau programme, sélectionner NOUV, taper ensuite un nom pour le programme (ALGO1 par exemple), valider avec Entrer. Il ne reste plus qu'à écrire le programme. Pour modifier un programme existant, choisir EDIT dans le menu prgm ; pour l'exécution, taper Entrer deux fois.

1) Entrées, affectations, sorties

	Algorithme	Basic Casio	Basic TI
Affectation : instruction qui place une valeur en mémoire.	A=A+1	A+1→A	A+1 $^{STO}\rightarrow$ A
Entrée(lecture) : affectation qui utilise la saisie de l'utilisateur.	Saisir A	?→A	Input A
Initialisation : première affectation pour les données non saisies par l'utilisateur.	A=0	0→A	0 $^{STO}\rightarrow$ A
Sortie(écriture) : instruction qui affiche une donnée.	Afficher A	A◢	Disp A

Avec ces quelques instructions, on peut écrire l'algorithme suivant qui calcule l'image y d'un nombre x par une fonction f (x étant une donnée saisie par l'utilisateur) et qui affiche ce nombre.

On va prendre $f(x)=\frac{1}{(5x-2)(2-x)}$: Saisir A ; B=1÷(5A-2)÷(2-A) ; Afficher B

Algorithme	Programme en Basic Casio	Programme en Basic TI
Saisir A B=1/(5A-2)(2-A) Afficher B	?→A 1÷(5A-2)÷(2-A)→B B◢	Input A 1÷(5A-2)÷(2-A)$^{STO}\rightarrow$B Disp B

Commentaire 1 : Vous avez noté l'inversion entre l'écriture dans l'algorithme B=1÷(5A-2)÷(2-A), qui affecte dans la mémoire B le résultat du calcul 1÷(5A-2)÷(2-A), et l'écriture dans les programmes 1÷(5A-2)÷(2-A)→B qui effectue la même chose (cette inversion n'est pas obligatoire dans tous les langages de programmation ; en Python par exemple, cette affectation s'écrit sans inversion B=1÷(5A-2)÷(2-A)).

Commentaire 2 : Ici, on peut utiliser une seule mémoire (il serait plus juste de dire une seule variable). On a choisit de prendre A, mais ça aurait pu être X ou B. L'algorithme « Saisir A ; A=1÷(5A-2)÷(2-A) ; Afficher A » est correct et on écrit dans le programme 1÷(5A-2)÷(2-A)→A (on utilise l'ancienne valeur de A pour calculer l'expression et on affecte le résultat à la mémoire A, écrasant l'ancienne valeur qui est alors perdue).

Commentaire 3 : Dans ce texte, j'utilise le séparateur d'instructions « ; » pour écrire notre algorithme sur une seule ligne, mais généralement, pour séparer les instruction, on va à la ligne sur la calculatrice (avec la touche « Enter »).

Commentaire 4 : L'affichage de B n'est pas indispensable car la calculatrice affiche, par défaut, la dernière valeur calculée. Ajouter l'instruction B◢ provoque un double affichage, ce qui fait que le résultat est affiché une 1ère fois avec la mention « Disp » (Display) qui indique qu'il s'agit d'un affichage demandé par le programme, et ensuite (après avoir relancé le programme avec la touche « Enter »), il est affiché une 2de fois, car le programme est terminé.

- Programmer cet algorithme sur votre calculatrice, puis tester le avec la valeur x=1 (on trouve $y=\frac{1}{3}\approx 0{,}333$). Compléter le tableau de données

x	0	0,3	0,39	0,5	1	1,5	1,9	2,1	2,5	3
$f(x)$	-0,25	-1,176...	-12,42	1,333...	0,333...	0,363...	1,333...	-1,176...	-0,19	-0,08

Voilà. J'ai ajouté quelques valeurs proches des valeurs interdites pour observer ce qui se passe quand on s'en approche...

Commentaire 5 : Rassurez-vous, votre calculatrice dispose de fonctions plus performantes pour obtenir un tableau de données (et aussi l'affichage graphique). Nous verrons cela dans un prochain TD... Ce premier exercice est juste destiné à se familiariser avec le mode programme, l'affectation des valeurs dans les mémoires.

2) Boucles

	Algorithme	Basic Casio	Basic TI
Boucle « pour » : on veut faire un certain nombre (N) de tours	Pour I allant de 1 à N : ….	For 1→I To N .... Next	For (I,1,N) ... End
Boucle « tant que » : on fait des tours tant qu'une condition (I<N) est vraie.	Tant que I<N : ….	While I<N …. WEnd	While I<N …. End

a) Voici un algorithme qui calcule la somme s des n premiers carrés d'entiers : $s=1^2+2^2+3^2+...+n^2$ (n étant une donnée saisie par l'utilisateur) et qui l'affiche : Saisir N ; S=0 ; Pour I allant de 1 à N : S=S+I^2 ; Afficher S

- Programmer cet algorithme sur votre calculatrice. Tester le programme avec la valeur n=5 (on trouve s=55).

Algorithme	Programme en Basic Casio	Programme en Basic TI
Saisir N S=0 Pour I allant de 1 à N S=S+I² fin de la boucle « Pour » Afficher S	?→N 0→S For 1→I To N S+I^2→S Next S◢	Input N 0 STO→S For (I,1,N) S+I^2 STO→S End Disp S

Tester le programme avec la valeur n=5 (on trouve s=55).
Calculer s pour n=100. On trouve s=338 350.

Commentaire 1 : Une boucle « pour » utilise toujours une variable de contrôle, appelée un compteur, qui compte le nombre de tours de boucle en incrémentant (augmentant) à chaque passage, cette valeur de 1. Le programmeur n'a pas besoin de faire cette incrémentation, c'est le programme qui s'en charge. Certains langages de programmation permettent de modifier cette valeur de l'incrément (qui est par défaut de 1) mais pas les langages Basic des calculatrices utilisées.

Commentaire 2 : Lorsqu'on écrit une boucle « pour », on peut tester par précaution la valeur qu'afficherait le programme en prenant la valeur la plus petite possible de la variable qui provoque la sortie de la boucle (ici c'est N). En entrant N=0, on n'entre pas dans la boucle. On affichera S=0 ce qui correspond bien à $s=0^2$.
En entrant N=1, on entre une fois dans la boucle avec I=1. On calcule $S=0+1^2=1$ et on affiche S=1 ce qui correspond bien à $s=0^2+1^2$.

b) Écrire un algorithme qui détermine la valeur de n à partir de laquelle la somme $s=1^2+2^2+...+n^2$ dépasse un nombre m donné (m étant une donnée saisie par l'utilisateur) et qui affiche cette valeur de n.

- Programmer cet algorithme sur votre calculatrice (on peut modifier le programme précédent ou en écrire un nouveau). Tester le programme avec la valeur m= 50 (on trouve n=5). Calculer n pour m= 10^6 : n=.....

Commentaire 1 : Ici, on ne sait pas combien de tours de boucle il faut faire. On doit donc utiliser une boucle « tant que » avec une condition (un test) qui est vraie au départ et qui, lorsqu'elle deviendra fausse, provoquera la sortie de boucle. Ici, la condition est S<M. Celle-ci est réalisée (vraie) dès lors qu'on entre une valeur de M>0. Cela n'a pas de sens de s'interroger sur ce qui se passerait si on entrait une valeur M≤0. Donc on entre dans la boucle, et au fur et à mesure que S est augmenté, on teste si la valeur de S vérifie toujours la condition. Lorsque S a dépassé M, on sort de la boucle et on affiche la valeur de I qui a provoqué cette sortie.

Algorithme	Programme en Basic Casio	Programme en Basic TI
Saisir M I=0 S=0 Tant que S<M I=1+I S=S+I² fin de la boucle « tant que » Afficher I Afficher S	?→M 0→I 0→S While S<M 1+I→I S+I^2→S WhileEnd (ou WEnd) I◢ S◢	Input N 0 STO→I 0 STO→S While S<M 1+I STO→I S+I^2 STO→S End Disp I Disp S

Commentaire 2 : Dans l'algorithme d'une boucle « tant que », on doit gérer l'initialisation et l'incrémentation du compteur I puisque la boucle « tant que » fonctionne sans compteur. Ce qu'il ne faut pas faire, c'est inverser, sans réfléchir, les instructions « I=1+I puis S=S+I² » en écrivant « S=S+I² puis I=1+I », ou alors il faudrait initialiser I à 1 et aussi, il faudrait aussi enlever 1 à la valeur finale affichée. L'ordre des instructions a une importance ici car on utilise I pour calculer S.

Commentaire 3 : On aurait pu appeler le compteur N au lieu de I puisque ici, on n'utilise pas ce nom de variable. La notation utilisée pour la puissance 2 n'est peut-être pas celle que vous avez utilisé (I² est possible sur les calculatrice) mais c'est la notation en ligne généralement admise des puissances. En Python, on notera I**2.

Tester le programme avec les valeurs m= 50 (on trouve n=5 et s=55). Calculer n pour m= 10^6 : On trouve n=144 et s=1 005 720.

Notre programme affichant S, on peut le donner. Mais, dans la consigne, ce n'était pas demandé.

3) Instructions conditionnelles

	Algorithme	Basic Casio	Basic TI
Si une condition (I<N) est vraie on exécute les instructions I_1, sinon (facultatif) on exécute les instructions I_2.	Si I<N alors I_1 sinon I_2	IF I<N Then I_1 Else I_2 IfEnd	IF I<N Then I_1 ELSE I_2 End

Écrire un algorithme qui demande d'entrer un nombre x (entre 0 et 21) et qui calcule l'aire du polygone restant (situation vue en cours). Rappel $A(x)=441-2x^2$ si $x\leq 10,5$ et $A(x)=441-2x^2+4(10,5-x)^2$ si $x\geq 10,5$.

- Programmer cet algorithme sur votre calculatrice (on peut modifier le programme 1).
  Compléter le tableau de données de la fonction A.

x	0	2	4	6	8	10	10,5	11	13	15	17	19	21
$A(x)$													

Algorithme	Programme en Basic Casio	Programme en Basic TI
Saisir A B=441-2A² Si A≥10,5 Alors B=B+4(10,5-A)² fin du Si Afficher B	?→A 441-2A²→B If A≥10,5 Then B+4(10,5-A)²→B IfEnd B◢	Input A 441-2A²→B If A≥10,5 Then B+4(10,5-A)²→B End Disp B

Commentaire 1 : On peut faire autrement, bien sûr, sans changer le résultat. Par exemple, voici un autre algorithme qui utilise la syntaxe complète : Si …. Alors..... Sinon.....

Algorithme *bis*	Programme *bis* en Basic Casio	Programme *bis* en Basic TI
Saisir A Si A<10,5 Alors A=441-2A² Sinon A=441-2A²+4(10,5-A)² fin du Si Afficher A	?→A If A<10,5 Then 441-2A²→A Else 441-2A²+4(10,5-A)²→A IfEnd A◢	Input A If A<10,5 Then 441-2A²→A Else 441-2A²+4(10,5-A)²→A End Disp A

Commentaire 2 : Ici, la mémoire A suffit pour faire le travail puisque le test de la valeur de A est effectué avant le calcul de son image. Bien sûr, on pourrait utiliser B comme précédemment pour mettre l'image de A (mais, par contre, on ne pourrait plus faire comme précédemment B=B+4(10,5-A)²).