Calculs de termes d'une suite, de somme de termes, de seuil

Auteur : Thomas Léchenne

Niveaux scolaires :
Première

Mots clés :
Algorithme/Programmation
Suites

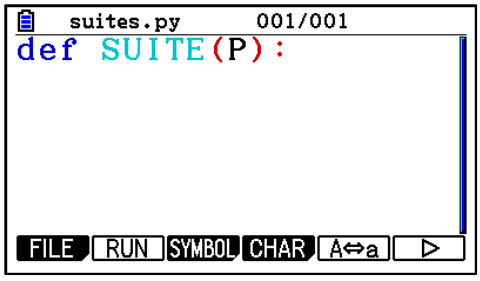
ENONCE

On considère la suite (u_n) définit par $u_0 = 1$ et par la relation de récurrence, $\forall n \in \mathbb{N}$ $u_{n+1} = \frac{3}{2}u_n - 1$.

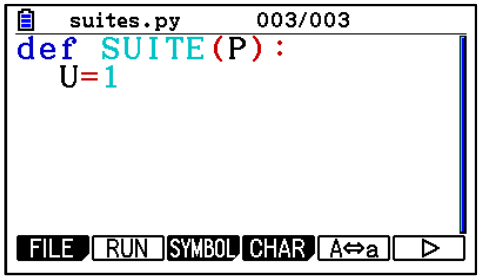
Ecrire, en langage Python, un algorithme donnant la valeur de u_p, la valeur de l'entier naturel p étant donnée.

Utiliser ce dernier pour construire un programme donnant la somme des termes jusqu'à u_k, la valeur de l'entier naturel k étant donnée ainsi qu'un autre programme donnant la plus petite valeur de n telle que $u_n < A$, la valeur du réel A étant elle aussi donnée.

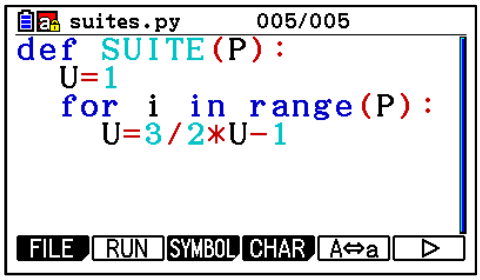
Commençons par nommer notre fonction "SUITE" à l'aide de la commande def (disponible dans le catalogue en appuyant sur les touches SHIFT 4). Elle aura un argument qui sera l'indice du terme voulu, P (attention à ne pas oublier les ":" pour démarrer l'indentation).



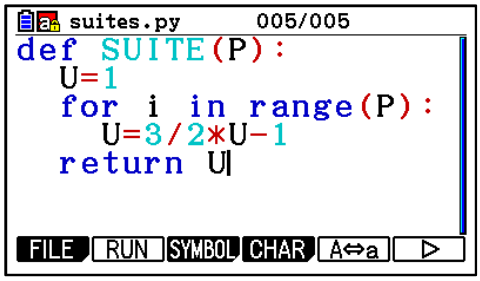
Ensuite, on initialise la variable U, variable qui va prendre les valeurs successives de la suite (u_n). Cette variable prend la valeur 1 au départ.



Pour continuer, nous allons calculer les termes suivants, un par un, en utilisant la relation de récurrence, et ce, à l'aide d'une boucle Pour (A noter que la variable i prend successivement les valeurs 0,…, P-1). Attention à ne pas oublier d'écrire le symbole de multiplication. Python considère que 2U est une variable à part entière alors que 2*U est deux fois la variable U.

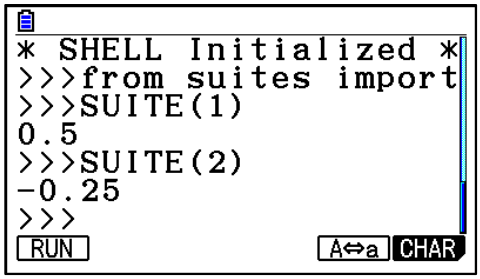


Pas besoin d'instruction de sortie de boucle : il suffit de sortir de l'indentation (en supprimant les deux espaces qui la symbolise) pour sortir de la boucle.
Il ne reste plus qu'à renvoyer la valeur de U.

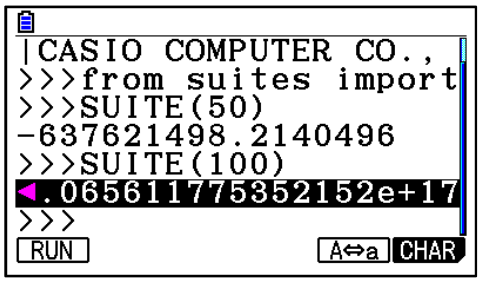


On peut tester le programme en pressant la touche F2 **{Run}** (on enregistre au passage). On vérifie que les premiers termes sont bons en comparant avec le calcul "à la main" :

$(u_1 = \frac{3}{2}u_0 - 1 = \frac{1}{2}$ et $u_2 = \frac{3}{2}u_1 - 1 = -\frac{1}{4})$



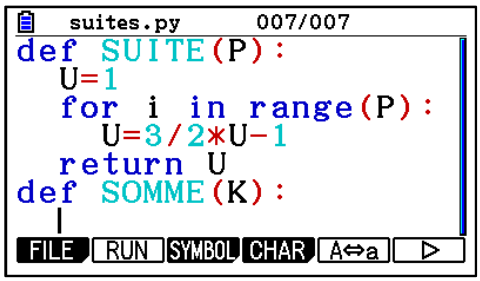
Et on peut ensuite calculer des termes un peu plus "lointains" pour avoir, par exemple, une idée de la limite (si celle-ci existe).
Ainsi, on peut conjecturer que $\lim_{n\to\infty} u_n = -\infty$.



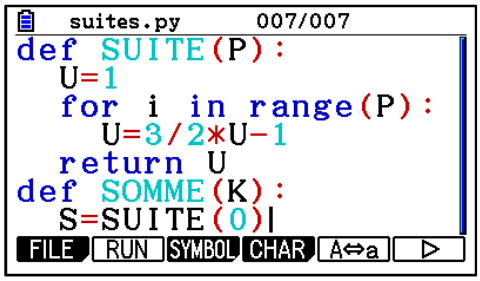
Algorithme de somme :

Pour cet algorithme, nous allons réutiliser la fonction SUITE programmée ci-dessus et faire la somme des termes un à un. On reste donc dans le même fichier pour que les deux fonctions soient chargées en même temps par la console d'exécution (on supprime cependant l'indentation pour créer une nouvelle fonction).

On crée une fonction SOMME qui aura pour argument l'indice k du dernier terme à sommer.

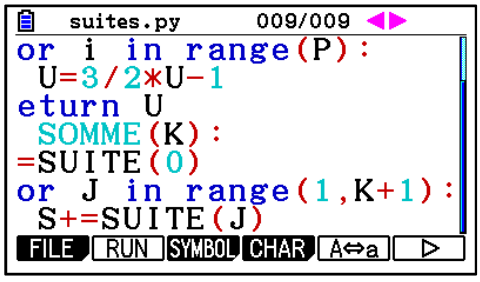


On initialise la variable S qui va stocker les valeurs successives de la somme. Cette somme commence donc par la valeur de u_0, c'est-à-dire à SUITE(0).

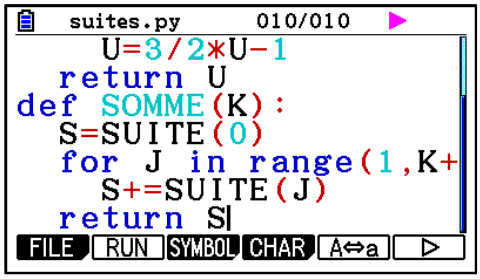


Ensuite, on ajoute les termes de la suite un à un à notre variable S. On fait donc une boucle Pour (pour J allant de 1 à K, ce qui se note "for J in range(1,K+1) ").

La notation "+=" signifie que l'on ajoute à notre variable S ce qui est à droite du signe égal. On aurait pu écrire plus simplement (mais moins rapidement) S=S+SUITE(J).

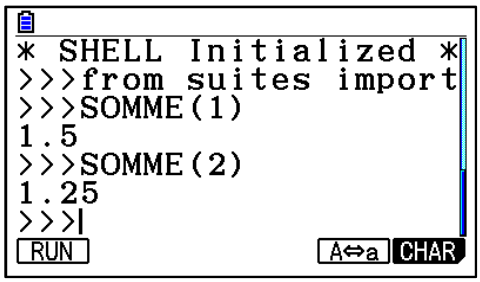


Il ne reste plus alors qu'à renvoyer la valeur de la somme S.



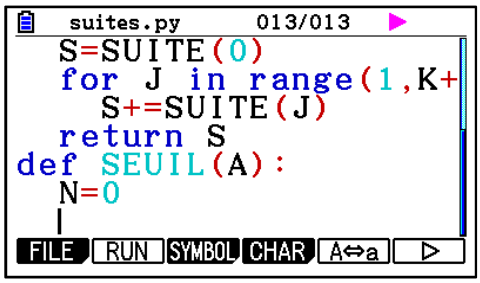
On teste l'algorithme avec les premières valeurs de la somme pour vérifier qu'il n'y a pas de problème. A la main, on a :

$$\sum_{i=0}^{1} u_i = u_0 + u_1 = \frac{3}{2} \; et \sum_{i=0}^{2} u_i = u_0 + u_1 + u_2 = \frac{5}{4}$$

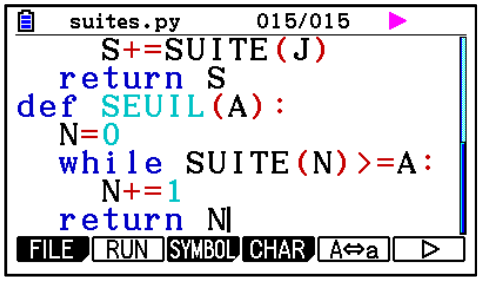


Algorithme de seuil :

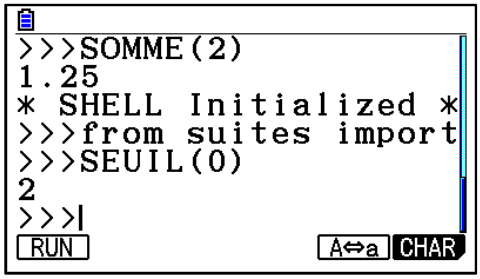
Une fois de plus on va réutiliser la fonction SUITE. On poursuit donc l'écriture dans le même fichier et on crée cette fois une fonction SEUIL qui aura pour argument la valeur réelle A. Dans cette fonction on initialise l'indice N du terme de la suite (on l'initialise à 0).



On va ensuite tester si le terme de la suite d'indice N est strictement plus petit que A et on continue tant que cette condition n'est pas réalisée (tant que $u_n \geq A$). On incrémente d'un la variable N à l'aide de la notation "+="
Au final, il ne reste plus qu'à renvoyer la dernière valeur de N.



On peut ensuite tester notre algorithme (on sait que la première valeur de n pour laquelle $u_n < 0$ est 2 puisque u_2 est la première valeur strictement négative de la suite).



En admettant que notre conjecture sur la limite est bonne, on peut chercher la première valeur de n pour laquelle $u_n < -10^6$ (Rappel : en Python l'exposant se note **, la calculatrice traduit automatiquement la touche puissance).
Ainsi (u_n) est strictement inférieure à $-1\,000\,000$ à partir de $n = 35$.