I. Suites

On appelle suite toute fonction de IN vers IR, qui à un nombre n associe son image u_n, appelé **terme général** de la suite.

On peut la définir (c'est-à-dire permettre de déterminer les termes u_1, u_2, u_3 ... de deux façons différentes :

→ À la façon d'une fonction, en donnant un moyen de calculer directement u_n à partir de n.

On a une formule de u_n en fonction de n (type F1 sur mode récurrence de la calculatrice)

Exemples : $\boldsymbol{u_n = \frac{1}{n}}$; $\boldsymbol{u_1 = \frac{1}{1} = 1}$; $\boldsymbol{u_2 = \frac{1}{2} = 0,5}$; $\boldsymbol{u_3 = \frac{1}{3}}$; ...

$\boldsymbol{v_n = 5n - 2}$; $\boldsymbol{v_0 = -2}$; $\boldsymbol{v_1 = 3}$; $\boldsymbol{v_2 = 8}$; $\boldsymbol{v_3 = 13}$

→ Par **récurrence**, en donnant $\begin{cases} \text{Le premier terme } u_0 \\ \text{La relation qui relie un terme } u_n \text{ à son suivant } u_{n+1} \end{cases}$

Exemple : $\begin{cases} u_0 = 1 \\ u_{n+1} = 2u_n + 1 \end{cases}$

$u_1 = 2u_0 + 1 = 2\times1 + 1 = 3$;
$u_2 = 2u_1 + 1 = 2\times3 + 1 = 7$

```
saisir n
u prend la valeur 1
Pour i allant de 1 à n
u prend la valeur 2u+1
FinPour
Afficher u
```

Programme pour calculer le terme de rang n après saisie de n

Algorithme	*Programme casio*
Saisir N Pour K allant de 1 à N U prend la valeur [*expression de la suite*] FinPour Afficher U	0→N "RANG="?→N For 1→K to N [*expression de la suite*] → U ***Next***

II. Algorithmique

On cherche à déterminer tous les termes d'une suite (définie en fonction de n) jusqu'à un certain rang P.

Algorithme	*Programme casio*
N prend la valeur 0 Saisir P Tant que N ≤ P U prend la valeur [*expression de la suite*] Afficher U N prend la valeur N+1 Fin de boucle.	0→N "RANG="?→P While N ≤ P [*expression de la suite*] → U U ▲ N+1 → N WhileEnd

III. Limite d'une suite

a. Limite infinie

Soit (u_n) une suite

Si pour tout entier naturel p, on peut trouver un rang à partir duquel tous les termes (u_n) sont supérieurs à 10^p, alors on dit que la suite (u_n) a pour limite +∞ quand n tend vers +∞.

On note : $\lim\limits_{n \to +\infty} u_n = +\infty$

Concrètement, on mettra en évidence cette limite en montrant qu'on peut rendre u_n « aussi grand qu'on veut » à l'aide d'un programme donc voici l'algorithme :

Algorithme	*Programme casio*
N prend la valeur 0	0→N
U prend la valeur 0	0→U
Saisir P	"RANG="?→P
Tant que U est inférieur ou égal à 10^P	While U ≤ 10^P
N prend la valeur N+1	N+1 → N
U prend la valeur [*expression de la suite*]	[*expression de la suite*] → U
Fin tant que.	Whileend
Afficher N	N

b. Limite finie

Rappel : le nombre |a – b| se lit « valeur absolue de a – b » et est égal à la distance entre les nombres a et b, il est donc toujours positif.

Soit (u_n) une suite et l un nombre donné.

Si pour tout entier naturel p, on peut trouver un rang à partir duquel tous les termes (u_n) sont à une distance de l inférieure à 10^{-p}, alors on dit que la suite u_n a pour limite l quand n tend vers +∞. On note : $\lim\limits_{n \to +\infty} u_n = l$

Concrètement, on mettra en évidence cette limite en montrant qu'on peut rendre u_n « aussi proche de l qu'on veut » à l'aide d'un programme donc voici l'algorithme :

Algorithme	*Programme casio*
N prend la valeur 0	0→N
U prend la valeur [u_0]	[u_0] →U
Saisir L	"10^-P P="?→P
Saisir P	"LIMITE="?→L
Tant que |U-L| est supérieur ou égal à 10^(-P)	While abs(U-L) ≥ 10^P
N prend la valeur N+1	N+1 → N
U prend la valeur [*expression de la suite*]	[*expression de la suite*] → U
Fin tant que.	whileend
Afficher N	N