Savoir DÉTERMINER DES TERMES D'UNE SUITE AU MOYEN D'UN OUTIL INFORMATIQUE

Les exercices **1.** *à* **4.** *travaillent toutes les techniques de base :*

le **1.** *utilise une formule explicite, le* **2.** *utilise une relation de récurrence, le* **3.** *utilise une formule mixte et le* **4.** *présente une difficulté de compréhension au niveau du test de l'algorithme.*

Les autres sont des extraits d'exercices posés au Baccalauréat.

Les deux derniers sont difficiles : le **9.** *propose deux suites croisées, et le* **10.** *utilise une relation de récurrence double.*

Vous devez savoir :

- Utiliser **la calculatrice** pour calculer des termes lointains (sans justification). Attention à bien distinguer les manières d'entrer la suite lorsqu'elle est définie par une formule explicite, par une relation de récurrence, ou par une formule mixte.
- Rédiger un algorithme pour :
    - **afficher un terme de rang N donné** :

Suite explicite avec formule $u_n = f(n)$ et premier rang 0 (parfois 1, attention)

Pas besoin d'algorithme

car on a accès directement à u_N en calculant $f(N)$

Suite récurrente avec relation $u_{n+1} = f(u_n)$ et premier terme u_0 (parfois u_1, attention)

Boucle *Tant que...*

```
n ← 0
u ← premier terme u0
Tant que n < N faire
        n ← n + 1
        u ← f(u)
Fin du Tant que
Afficher u
```

Passages dans la boucle	n passe de	u passe de
Premier	0 à 1	u_0 à $f(u_0) = u_1$
Deuxième	1 à 2	u_1 à $f(u_1) = u_2$
⋮	⋮	⋮
Dernier	N-1 à N	u_{N-1} à $f(u_{N-1}) = u_N$

Après le dernier passage, on ne peut plus entrer dans la boucle car n contient N et il ne vérifie plus le test $n < N$.

On sort donc du dernier passage dans la boucle avec u qui contient u_N et que nous devons afficher.

Boucle *Pour...*

```
u ← premier terme u0
Pour i allant de 1 à N faire
        u ← f(u)
Fin du Pour
Afficher u
```

→ i sert à compter le nombre de fois qu'on doit passer d'un terme au suivant.

→ Après le 1er passage ($i = 1$), u est passé de u_0 à $u_1 = f(u_0)$.

→ Après le $N^{\text{ème}}$ passage ($i = N$), u contient u_N que nous devons afficher.

- **afficher <u>tous les termes jusqu'à</u> un rang N donné** :

Suite explicite avec formule $u_n = f(n)$ et premier rang 0 (parfois 1, attention)

Boucle *Tant que...*

$n \leftarrow 0$
$u \leftarrow f(0)$ → C'est u_0 .
Afficher u → u_0 affiché (ne pas l'oublier car il ne sera pas affiché dans les boucles).
Tant que $n < N$ faire
    $n \leftarrow n + 1$
    $u \leftarrow f(n)$
    Afficher u
Fin du Tant que

Passages dans la boucle	n passe de	u prend la valeur	
Premier	0 à 1	$u_1 = f(1)$	u_1 affiché
Deuxième	1 à 2	$u_2 = f(2)$	u_2 affiché
⋮	⋮	⋮	⋮
Dernier	N-1 à N	$u_N = f(N)$	u_N affiché

Après le dernier passage dans la boucle, on ne peut plus y passer car n contient N et il ne vérifie plus le test $n < N$.

Boucle *Pour...*

Pour n allant de 0 à N faire → Le rang n sert de compteur !
    $u \leftarrow f(n)$
    Afficher u
Fin du Pour

Passages dans la boucle	u prend la valeur	
$n = 0$	$f(0) = u_0$	affiché
$n = 1$	$f(1) = u_1$	affiché
⋮	⋮	⋮
$i = N$	$u_N = f(N)$	affiché

C'est très astucieux, on se sert de l'incrémentation automatique du Pour pour passer de n à n+1 .

Suite récurrente avec relation $u_{n+1} = f(u_n)$ et premier terme u_0 (parfois u_1 , attention)

Boucle *Tant que...*

$n \leftarrow 0$
$u \leftarrow$ premier terme u_0
Afficher u → u_0 affiché (ne pas l'oublier car il ne sera pas affiché dans les boucles).
Tant que $n < N$ faire
    $n \leftarrow n + 1$
    $u \leftarrow f(u)$
    Afficher u
Fin du Tant que

Passages dans la boucle	n passe de	u passe de	
Premier	0 à 1	u_0 à $f(u_0) = u_1$	u_1 affiché
Deuxième	1 à 2	u_1 à $f(u_1) = u_2$	u_2 affiché
⋮	⋮	⋮	⋮
Dernier	N-1 à N	u_{N-1} à $f(u_{N-1}) = u_N$	u_N affiché

Boucle *Pour...* (version 1)

$u \leftarrow$ premier terme u_0
Afficher u → u_0 affiché.
Pour i allant de 1 à N faire
    $u \leftarrow f(u)$
    Afficher u
Fin du Pour

Passages dans la boucle	u prend la valeur	
$i = 1$	$f(u_0) = u_1$	u_1 affiché
$i = 2$	$f(u_1) = u_2$	u_2 affiché
⋮	⋮	⋮
$i = N$	$f(u_{N-1}) = u_N$	u_N affiché

Boucle *Pour...* (version 2)

$u \leftarrow$ premier terme u_0
Pour n allant de 0 à N faire → Le rang n sert de compteur !
    Afficher u → L'astuce est de commencer par l'affichage, puis faire le calcul.
    $u \leftarrow f(u)$
Fin du Pour

Passages dans la boucle		u prend la valeur
$n = 0$	u_0 affiché	$f(u_0) = u_1$
$n = 1$	u_1 affiché	$f(u_1) = u_2$
$n = 2$	u_2 affiché	$f(u_2) = u_3$
⋮	⋮	⋮
$n = N - 1$	u_{N-1} affiché	$f(u_{N-1}) = u_N$
$n = N$	u_N affiché	$f(u_N) = u_{N+1}$

<u>Remarque</u> : Dans la dernière boucle, u_{N+1} est calculé pour rien.

- **afficher le <u>premier terme</u> dépassant une valeur A donnée** :

Suite explicite avec formule $u_n = f(n)$ et premier rang 0 (parfois 1, attention)

$n \leftarrow 0$
$u \leftarrow f(0)$
Tant que $u \leqslant A$ faire
    $n \leftarrow n+1$
    $u \leftarrow f(n)$
Fin du Tant que
Afficher u → Le dernier u sortant de la boucle est le premier terme dépassant A .
C'est celui que nous devons afficher.

Pas de variante avec boucle *Pour...*

Suite récurrente avec relation $u_{n+1} = f(u_n)$ et premier terme u_0 (parfois u_1 , attention)

$u \leftarrow$ premier terme u_0
Tant que $u \leqslant A$ faire
    $u \leftarrow f(u)$ → Les rangs ne sont d'aucune utilité... pas de n ...
Fin du Tant que
Afficher u → Le dernier u sortant de la boucle est le premier terme dépassant A .
C'est celui que nous devons afficher.

Pas de variante avec boucle *Pour...*

- **afficher le <u>premier rang</u> dont le terme dépasse une valeur A donnée** :

Suite explicite avec formule $u_n = f(n)$ et premier rang 0 (parfois 1, attention)

$n \leftarrow 0$
$u \leftarrow f(0)$
Tant que $u \leqslant A$ faire
    $n \leftarrow n+1$
    $u \leftarrow f(n)$
Fin du Tant que
Afficher n → Le dernier n sortant de la boucle est le rang du premier terme dépassant A .
C'est celui que nous devons afficher.

Pas de variante avec boucle *Pour...*

Suite récurrente avec relation $u_{n+1} = f(u_n)$ et premier terme u_0 (parfois u_1 , attention)

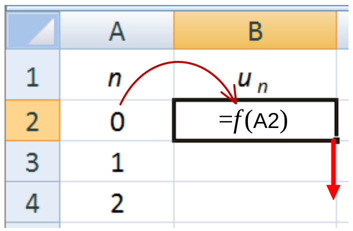
$n \leftarrow 0$
$u \leftarrow$ premier terme u_0
Tant que $u \leqslant A$ faire
    $n \leftarrow n+1$ → Les rangs ne sont d'aucune utilité... mais il faut les calculer pour afficher le dernier !
    $u \leftarrow f(u)$
Fin du Tant que
Afficher u → Le dernier n sortant de la boucle est le rang du premier terme dépassant A .
C'est celui que nous devons afficher.

Pas de variante avec boucle *Pour...*

- *Donner la formule à écrire dans une cellule de tableur pour calculer les termes d'une suite :*

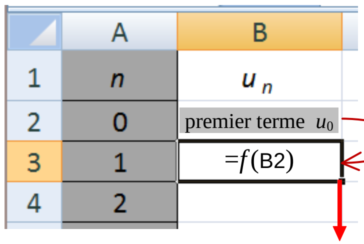
Suite explicite avec formule $u_n = f(n)$
***et premier rang* 0** (parfois **1**)

On passe des rangs aux termes :
on remplace n par A2 dans la formule.



Suite récurrente avec relation $u_{n+1} = f(u_n)$
et premier terme u_0 (parfois u_1)

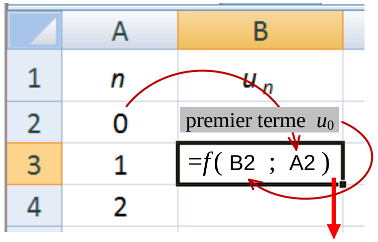
La colonne des rangs
ne sert pas au calcul
des termes. elle sert juste
d'indicateurs...



On passe d'un terme
au suivant :
on remplace u_n par
B2 dans la formule.

Suite définie par une formule mixte : $u_{n+1} = f(\,u_n\,;\,n\,)$
(u_{n+1} est fonction de u_n et de n)
et premier terme u_0 (parfois u_1)

On remplace u_n par B2 ...



... et n par A2
dans la formule.

1. On définit la suite (a_n) par $a_n = \dfrac{n^3 + 2n}{n + 1}$ pour tout $n \in \mathbb{N}$.

a) On a calculé ci-dessous les premiers termes de cette suite au moyen d'un tableur :

	A	B
1	Rangs	Termes
2	0	0
3	1	1,5
4	2	4
5	3	8,25

Proposer une formule à écrire dans la cellule B2 pour calculer a_0, sachant que cette formule « tirée vers le bas » dans la colonne devra permettre de calculer les termes successifs de la suite (a_n).

b) En observant les valeurs données par ce tableur ci-dessous, on conjecture que les termes a_n tendent vers $+\infty$ lorsque n tend vers $+\infty$.

	A	B
101	99	9704,97
102	100	9902,9703
103	101	10102,9706
104	102	10304,9709
105	103	10508,9712

Proposer un algorithme permettant de trouver le rang à partir duquel tous les termes a_n sont strictement supérieurs à 100 000 .

c) Programmer cet algorithme sur calculatrice pour trouver ce rang.

---

2. On définit la suite (u_n) par $\begin{cases} u_0 = 0{,}001 \\ u_{n+1} = 2\,(u_n + 1)\,. \end{cases}$

a) On a calculé ci-dessous les premiers termes à l'aide d'un tableur.

	A	B
1	Rangs	Termes
2	0	0,001
3	1	2,002
4	2	6,004
5	3	14,008

Quelles formules ont été entrées dans les cellules B2 et B3 pour pouvoir calculer tous les termes de cette suite en tirant une cellule vers le bas.

b) Proposer un algorithme permettant d'afficher les huit premiers termes de cette suite.

c) Proposer un algorithme permettant d'afficher le terme de rang 10.

d) Proposer un algorithme permettant d'afficher le rang du premier terme qui dépasse 10 000.

---

3. Soit la suite (u_n) définie par son premier terme $u_1 = \dfrac{3}{2}$ et la relation de récurrence $u_{n+1} = \dfrac{n\,u_n}{n + 1}$.

a) L'extrait de feuille de tableur ci-dessous montre les premiers termes de cette suite :

	A	B
1	n	u_n
2	1	1,5
3	2	0,75
4	3	0,5
5	4	0,375
6	5	0,3

Proposer une formule écrite dans la cellule B3 permettant de calculer les autres termes de la suite.

b) Proposer un algorithme permettant d'afficher le terme de rang 20.

c) Proposer un algorithme permettant d'afficher les dix premiers termes.

---

4. On définit la suite (v_n) par $v_n = \dfrac{2}{n-1} + 3$ pour tout $n \geqslant 2$.

a) On a calculé ci-dessous les premiers termes de cette suite au moyen d'un tableur :

	A	B
1	Rangs	Termes
2	2	5
3	3	4
4	4	3,66666667
5	5	3,5

Proposer une formule à écrire dans la cellule B2 pour calculer v_2 , sachant que cette formule « tirée vers le bas » dans la colonne devra permettre de calculer les termes successifs de la suite (v_n) .

b) En observant les valeurs données par ce tableur ci-dessous, conjecturer la limite de la suite.

	A	B
99	99	3,02040816
100	100	3,02020202
101	101	3,02
102	102	3,01980198
103	103	3,01960784

c) On propose l'algorithme suivant :

Variables : n entier naturel
v réel
Affecter à n la valeur 2
Affecter à v la valeur $\dfrac{2}{n-1} + 3$
Tant que $|v-3| \geqslant 0{,}01$ faire
$n \leftarrow n+1$
$v \leftarrow \dfrac{2}{n-1} + 3$
Fin du Tant que
Afficher n

Expliquer quoi sert cet algorithme.
Quelle valeur affiche-t-il ?

---

5. On considère la suite (u_n) définie par $u_0 = 1$ et, pour tout entier naturel n , $u_{n+1} = \sqrt{2u_n}$.

On considère l'algorithme suivant :

Variables : n est un entier naturel
u est un réel positif
Initialisation : Demander la valeur de n
Affecter à u la valeur 1
Traitement : Pour i variant de 1 à n :
| Affecter à u la valeur $\sqrt{2u}$
Fin de Pour
Sortie : Afficher u

a) Donner une valeur approchée à 10^{-3} près du résultat qu'affiche cet algorithme lorsque l'on choisit $n = 3$.

b) Que permet de calculer cet algorithme ?

c) Recopier et compléter le tableau ci-dessous donnant des valeurs approchées par défaut à 10^{-5} près obtenues à l'aide de cet algorithme pour certaines valeurs de n :

n	15	16	17	18
Valeurs affichées				

Quelles conjectures peut-on émettre concernant la suite (u_n) ?

d) Dans cette question, on suppose que la suite (u_n) est croissante et admet 2 pour limite.
Recopier l'algorithme ci-dessous et le compléter par les instructions du traitement et de la sortie, de façon à afficher en sortie la plus petite valeur de n telle que $u_n > 1{,}999$.

Variables :	*n* est un entier naturel
	u est un réel
Initialisation :	Affecter à *n* la valeur 0
	Affecter à *u* la valeur 1
Traitement :	
Sortie :	

D'après Baccalauréat Amérique du Nord 2013

---

6. L'objet de cet exercice est l'étude de la suite (u_n) définie par son premier terme $u_1 = \dfrac{3}{2}$ et la relation de récurrence $u_{n+1} = \dfrac{nu_n + 1}{2(n+1)}$.

Pour calculer et afficher le terme u_9 de la suite, un élève propose l'algorithme ci-dessous :

Variables	*n* est un entier naturel
	u est un réel
Initialisation	Affecter à *n* la valeur 1
	Affecter à *u* la valeur 1,5
Traitement	Tant que *n* < 9
	Affecter à *u* la valeur ...
	Affecter à *n* la valeur ...
	Fin Tant que
Sortie	Afficher la variable *u*

Il a oublié de compléter deux lignes.

a) Recopier et compléter les deux lignes de l'algorithme où figurent des points de suspension.

b) Comment faudrait-il modifier cet algorithme pour qu'il calcule et affiche tous les termes de la suite de u_2 jusqu'à u_9 ?

c) Avec cet algorithme modifié, on a obtenu les résultats suivants, arrondis au dix-millième :

n	1	2	3	4	5	6	...	99	100
u_n	1,5	0,625	0,375	0,2656	0,2063	0,1693	...	0,0102	0,0101

Au vu de ces résultats, conjecturer le sens de variation et la limite de la suite (u_n) .

D'après Baccalauréat Centres Étrangers 2013

---

7. On considère la suite (u_n) définie par : $u_0 = 2$ et, pour tout entier naturel n : $u_{n+1} = \dfrac{1 + 0{,}5u_n}{0{,}5 + u_n}$.

a) On considère l'algorithme suivant :

Entrée	Soit un entier naturel non nul *n*
Initialisation	Affecter à *u* la valeur 2
Traitement et sortie	POUR *i* allant de 1 à *n* Affecter à *u* la valeur $\dfrac{1 + 0{,}5u}{0{,}5 + u}$ Afficher *u* FIN POUR

Reproduire et compléter le tableau ci-contre, en faisant fonctionner cet algorithme pour $n = 3$.
Les valeurs de u seront arrondies au millième.

i	1	2	3
u			

b) Pour $n = 12$, on a prolongé le tableau précédent et on a obtenu :

i	4	5	6	7	8	9	10	11	12
u	1,008 3	0,997 3	1,000 9	0,999 7	1,000 1	0,999 97	1,000 01	0,999 996	1,000 001

Conjecturer le comportement de la suite (u_n) à l'infini.

D'après Baccalauréat Asie 2013

---

8. On considère la suite numérique (v_n) définie pour tout entier naturel n par $\begin{cases} v_0 = 1 \\ v_{n+1} = \dfrac{9}{6 - v_n} \end{cases}$.

a) On souhaite écrire un algorithme affichant, pour un entier naturel n donné, tous les termes de la suite, du rang 0 au rang n .
Parmi les trois algorithmes suivants, un seul convient.
Préciser lequel en justifiant la réponse.

Algorithme N° 1

Variables :
v est un réel
i et *n* sont des entiers naturels
Début de l'algorithme :
Lire *n*
v prend la valeur 1
Pour *i* variant de 1 à *n* faire
    v prend la valeur 9/(6 – *v*)
Fin pour
Afficher *v*
Fin algorithme

Algorithme N° 2

Variables :
v est un réel
i et *n* sont des entiers naturels
Début de l'algorithme :
Lire *n*
Pour *i* variant de 1 à *n* faire
    v prend la valeur 1
    Afficher *v*
    v prend la valeur 9/(6 – *v*)
Fin pour
Fin algorithme

Algorithme N° 3

Variables :
v est un réel
i et *n* sont des entiers naturels
Début de l'algorithme :
Lire *n*
v prend la valeur 1
Pour *i* variant de 1 à *n* faire
    Afficher *v*
    v prend la valeur 9/(6 – *v*)
Fin pour
Afficher *v*
Fin algorithme

b) Pour $n = 10$, on obtient l'affichage suivant :

1	1,800	2,143	2,333	2,455	2,538	2,600	2,647	2,684	2,714

Pour $n = 100$, les derniers termes affichés sont :

2,967	2,968	2,968	2,968	2,969	2,969	2,969	2,970	2,970	2,970

Quelles conjectures peut-on émettre concernant la suite (v_n) ?

D'après Baccalauréat Liban 2013

9. Soient deux suites (u_n) et (v_n) définies par $u_0 = 2$ et $v_0 = 10$ et, pour tout entier naturel n,
$u_{n+1} = \dfrac{2u_n + v_n}{3}$ et $v_{n+1} = \dfrac{u_n + 3v_n}{4}$.

On considère l'algorithme ci-dessous :

```
Variables :  N est un entier
             U , V , W sont des réels
             K est un entier
Début :      Affecter 0 à K
             Affecter 2 à U
             Affecter 10 à V
             Saisir N
             Tant que K < N
                  Affecter K + 1 à K
                  Affecter U à W
                  Affecter (2U + V)/3 à U
                  Affecter (W + 3V)/4 à V
             Fin tant que
             Afficher U
             Afficher V
Fin
```

On exécute cet algorithme en saisissant $N = 2$.
Recopier et compléter le tableau donné ci-dessous donnant l'état des variables au cours de l'exécution de l'algorithme.

K	U	V	W
0			
1			
2			

D'après Baccalauréat Nouvelle Calédonie Novembre 2013

---

10. On considère la suite (u_n) définie par $\begin{cases} u_0 = 3 \\ u_1 = 8 \\ \text{pour tout entier } n \text{ supérieur ou égal à } 0,\ u_{n+2} = 5u_{n+1} - 6u_n. \end{cases}$

a) Calculer u_2 et u_3.

b) Pour tout entier naturel $n > 2$, on souhaite calculer u_n à l'aide de l'algorithme suivant :

```
Variables :        a, b et c sont des nombres réels
                   i et n sont des nombres entiers naturels supérieurs ou égaux à 2
Initialisation :   a prend la valeur 3
                   b prend la valeur 8
Traitement :       Saisir n
                   Pour i variant de 2 à n faire
                        c prend la valeur a
                        a prend la valeur b
                        b prend la valeur . . .
                   Fin Pour
Sortie :           Afficher b
```

Recopier la ligne de cet algorithme comportant des pointillés et les compléter.

D'après Baccalauréat Liban 2013 - Enseignement de spécialité