Histoire 1

- Archimède (vers 287 – 212 avant JC) est un mathématicien, physicien et inventeur de langue grecque qui vécut à Syracuse en Sicile. Dans *La méthode*, il démontre que l'aire $\mathscr{A}$ d'un disque est égale à l'aire $\mathscr{T}$ d'un triangle rectangle de hauteur le rayon R et de base la circonférence $2\pi R$. Il examine tous les cas possibles (méthode d'exhaustion) et prouve par l'absurde qu'on ne peut avoir ni $\mathscr{A} > \mathscr{T}$, ni $\mathscr{A} < \mathscr{T}$. Il s'appuie sur un axiome de continuité « *En la divisant successivement par* 2*, on peut rendre une quantité aussi petite que l'on veut* » et encadre l'aire du disque par celles de polygones inscrits et exinscrits avec un nombre croissants de côtés. Il obtient ainsi un encadrement de π remarquable pour l'époque $3+\dfrac{10}{71} < \pi < 3+\dfrac{1}{7}$. Ce principe sera repris bien plus tard pour approcher les nombres irrationnels par des suites.

- Héron d'Alexandrie (vers 170 – 117 avant JC) est un physicien et mathématicien grec, célèbre pour sa formule de l'aire d'un triangle de côtés a, b et c en fonction de son demi-périmètre p : $\mathscr{A} = \sqrt{p(p-a)(p-b)(p-c)}$. Son nom est associé à un algorithme d'approximation de $\sqrt{2}$, connu des Babyloniens 400 ans avant, noté actuellement sous la forme de la suite $r_0 = 2$ et $r_{n+1} = \dfrac{1}{2}\left(r_n + \dfrac{2}{r_n}\right)$.

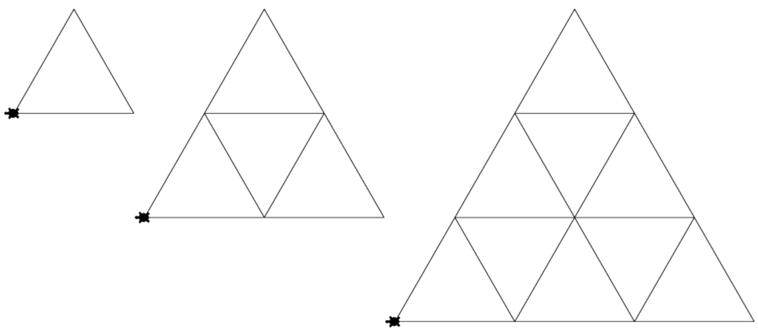
- Léonard de Pise, dit Fibonacci (1175 – 1240) est un mathématicien italien auteur du *Liber abaci* (1202), un recueil de problèmes algébriques, où il popularisa l'usage des chiffres arabes. Un énoncé est resté célèbre : « *Possédant au départ un couple de lapins, combien de couples de lapins obtient-on en douze mois si chaque couple engendre tous les mois un nouveau couple à compter du second mois de son existence ?* », il se modélise avec la suite $f_0 = f_1 = 1$ et $f_{n+2} = f_{n+1} + f_n$. On démontre que le rapport f_{n+1}/f_n tend vers le nombre d'Or $\dfrac{1+\sqrt{5}}{2}$ lorsque n tend vers $+\infty$.

1 Notion de suite

1.1 Construction de suites

Activité 1 *Châteaux de cartes*

On se propose de réaliser un château de cartes comme dans le graphique ci-dessous où on a représenté de gauche à droite des châteaux de un, deux ou trois étages. Les cartes sont exactement les côtés des petits triangles équilatéraux constituant les châteaux. Ainsi, il faut 3 cartes pour réaliser un château de un étage et 9 cartes pour un château de deux étages.



1. Soit un château de cartes. On numérote les étages de haut en bas en commençant à 1 et on appelle indice le numéro d'un étage. Compléter le tableau ci-dessous associant à chaque étage le nombre de cartes nécessaires pour le réaliser :

Étage d'indice	1	2	3	4	5	6	7
Nombre de cartes	3	6	9	...	...	...	...

2. Combien faut-il de cartes au total pour construire un château de 7 étages ? de 10 étages ?

3. On note $u(n)$ ou encore u_n, qui se lit *u indice n*, le nombre de cartes de l'étage d'indice n. De plus on note $v(n)$ ou v_n le nombre total de cartes d'un château de n étages.

Pour un château de 3 étages, on a donc :

$$u(1) = u_1 = 3,\ u(2) = u_2 = 6,\ u(3) = u_3 = 9 \text{ et } v(3) = v_3 = u_1 + u_2 + u_3 = 18.$$

a. Déterminer u_4, u_5, u_6, u_7 et v_4, v_5, v_6, v_7.

b. Exprimer u_n explicitement en fonction de n.

c. Établir une relation entre v_n, u_{n+1} et v_{n+1}.

d. Compléter l'algorithme et la fonction Python ci-dessous pour que la fonction chateau retourne le nombre de cartes nécessaires pour un château de n étages.

Algorithme

```
Fonction chateau(n):
  u ⟵ 0
  v ⟵ 0
  Pour k allant de 1 à n
    u ⟵ ...
    v ⟵ ...
  Retourne v
```

Python

```python
def chateau(n):
    u = 0
    v = 0
    for k in range(1, n+1):
        u = ...........
        v = ...........
    return v
```

1.2 Définition

Définition 1

Une suite est une fonction définie sur l'ensemble $\mathbb{N}$ des entiers naturels, les entiers positifs ou nuls.
Si u est le nom de la suite, l'image de n par u se note $u(n)$ (notation fonctionnelle) ou de manière plus usuelle u_n (notation indicielle). On l'appelle terme d'indice n ou de rang n ou **terme général** de la suite u. L'ensemble des termes de la suite se note alors $(u_n)_{n\in\mathbb{N}}$ ou (u_n) par abus de notation .

Capacité 1 *Maîtriser la notation indicielle, compléter une suite logique*

1. On considère la suites des entiers positifs impairs : $u_1 = 1$, $u_2 = 3$, ...

   Déterminer le terme d'indice 5 de cette suite. Quelle est la valeur de u_{12} ?

2. On considère la suite des décimales de $\pi \approx 3,14\ldots$ et on note $p_0 = 3$, $p_1 = 1$, $p_2 = 4$. Énumérer tous les termes de cette suite (p_n) dont vous pouvez déterminer la valeur exacte avec votre calculatrice.

   Lors du Pi Day du 14 mars 2019, une ingénieure japonaise a calculé 31 mille milliards de décimales de π en 121 jours avec la technologie Compute Engine de Google Cloud ...

3. Compléter la suite logique ci-dessous inventée par John Conway et surnommée *suite look and say* :

   $u_1 = 1$, $u_2 = 11$, $u_3 = 21$, $u_4 = 1211$, $u_5 = 111221$, $u_5 = \ldots$

1.3 Modèle d'évolution d'un phénomène discret

Activité 2 *Modèle d'évolution d'une population*

Une grande université, en pleine croissance d'effectifs, accueillait 27 500 étudiants en septembre 2016. Le président de l'université est inquiet car il sait que, malgré une gestion optimale des locaux et une répartition des étudiants sur les divers sites de son université, il ne pourra pas accueillir plus de 33 000 étudiants.
Une étude statistique lui permet d'élaborer un modèle de prévisions selon lequel, chaque année :

- 150 étudiants démissionnent en cours d'année universitaire (entre le 1$^{\text{er}}$ septembre et le 30 juin) ;
- les effectifs constatés à la rentrée de septembre connaissent une augmentation de 4 % par rapport à ceux du mois de juin qui précède.

Pour tout entier naturel n, on note u_n le nombre d'étudiants estimé selon ce modèle à la rentrée de septembre $2016 + n$. On a donc $u_0 = 27\,500$.

1. **a.** Déterminer le nombre d'étudiants en juin 2017.

   b. Justifier que le nombre d'étudiants à la rentrée de septembre 2017 était de 28 444.

2. **a.** Que représente u_3 ? Calculer sa valeur à la main.

   b. Justifier que, pour tout entier naturel n, on a $u_{n+1} = 1,04u_n - 156$.

   c. En programmant la suite (u_n) avec le mode Suite de la calculatrice (**tutoriel page 339 pour TI et page 349 pour Numworks**), déterminer le nombre d'étudiants prévu par ce modèle à la rentrée de septembre 2025.

3. **a.** Compléter la fonction `seuil` ci-dessous pour qu'elle retourne l'année à partir de laquelle le nombre d'étudiants à accueillir dépassera la capacité maximale de l'établissement.

Algorithme

```
Fonction seuil():
    N ⟵ 0
    U ⟵ 27 500
    Tant que U......faire
        N ⟵ N + 1
        U ⟵ ......
    Fin Tant que
    Retourne N + 2016
```

Python

```python
def seuil():
    n = 0
    u = 27500
    while u ............:
        n = n + 1
        u = .............
    return n + 2016
```

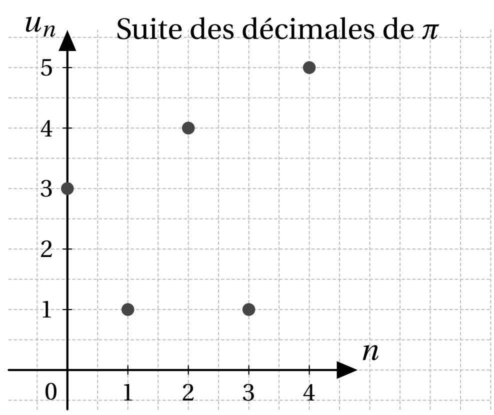
b. Quelle est la valeur retournée par l'appel de fonction `seuil()`, répondre en complétant un tableau d'évolution des variables `U` et `N`.

2 Différents modes de génération d'une suite

2.1 Suite définie par extension

Définition 2

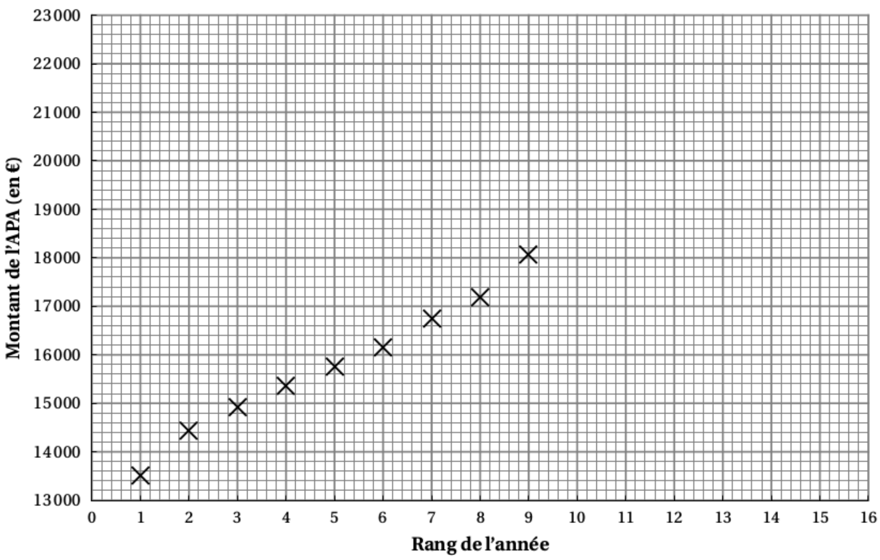
- Une suite est **définie par extension** si tous ses termes successifs nous sont donnés, comme par exemple une série statistique.
- Dans un repère, une suite (u_n) est représentée graphiquement par un **nuage de points** de coordonnées $(n; u_n)$.



Capacité 2 ***Utiliser plusieurs registres (graphique, algébrique) pour étudier une suite***

L'Allocation Personnalisée d'Autonomie (APA) est une allocation destinée aux personnes âgées de 60 ans et plus en perte d'autonomie.

Une série statistique nous donne le montant en euros de l'APA dans un département fixé depuis 2007. On note a_n le montant pour l'année $2006 + n$ et le nuage de points ci-dessous représente les neuf premiers termes de la suite (a_n).



1. **a.** Compléter le tableau ci-dessous par lecture graphique :

Indice n	1	2	3	4	5	6	7	8	9
a_n	13 504	...	14 914	15 351	15 751	16 144	16 744	...	18 070

   b. Quel était le montant de l'APA en 2013 ?

2. Le conseil départemental décide d'une augmentation de 5% par an à partir de 2015.

   a. Calculer a_{10} et compléter le nuage de points.

   b. Quelle formule faut-il saisir en `C2` dans la feuille de calcul ci-dessous pour calculer le montant de l'APA à partir de 2015 ?

	A	B	C	D	E	F
1	n	9	10	11	12	13
2	a_n	18 070	...	...	...	...

2.2 Suite définie par une formule de récurrence

Définition 3

- Une suite $(u_n)_{n\in\mathbb{N}}$ est définie par une **formule de récurrence** si son terme général u_n s'exprime en fonction de termes d'indices inférieurs.

- La suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 = 1$ et pour tout entier $n \geqslant 0$, $u_{n+1} = 2u_n + 1$ est un exemple de suite définie par une récurrence d'ordre 1.

  La même suite peut être définie par $u_0 = 1$ et pour tout entier $n \geqslant 1$, $u_n = 2u_{n-1} + 1$.

- La suite $\left(f_n\right)_{n\in\mathbb{N}}$ définie par $f_0 = f_1 = 1$ et pour tout entier $n \geqslant 0$, $f_{n+2} = f_{n+1} + f_n$ est un exemple de suite définie par une récurrence d'ordre 2.

  La même suite peut être définie par $f_0 = f_1 = 1$ et pour tout entier $n \geqslant 2$, $f_n = f_{n-1} + f_{n-2}$.

Capacité 3 ***Calculer des termes d'une suite définie par une relation de récurrence, voir exo 1 p. 11***

1. Soit la suite $(u_n)_{n\geqslant 0}$ définie par : $u_0 = 4$ et, pour tout entier naturel n, $u_{n+1} = -\frac{1}{2}u_n + 2$.

a. Détailler les calculs de u_1 et u_2.

b. Avec le mode `suite` ou `recurrence` de la calculatrice (**tutoriel page 339 pour TI et page 349 pour Numworks**), calculer une valeur décimale approchée à 10^{-6} près de u_{14}.

2. Soit la suite (u_n) définie par $u_0 = 2$ et pour tout entier naturel n, par $u_{n+1} = u_n + n^2 + 1$.

a. Détailler les calculs de u_1 et u_2.

b. Avec le mode `suite` ou `recurrence` de la calculatrice calculer u_{10}.

2.3 Suite définie par une formule explicite

Définition 4

- Une suite $(u_n)_{n\in\mathbb{N}}$ est définie par une **formule explicite** si son terme général u_n peut s'exprimer directement comme une fonction algébrique de n.
- Ainsi on peut calculer directement un terme d'indice quelconque d'une suite définie explicitement sans calculer d'autres termes.

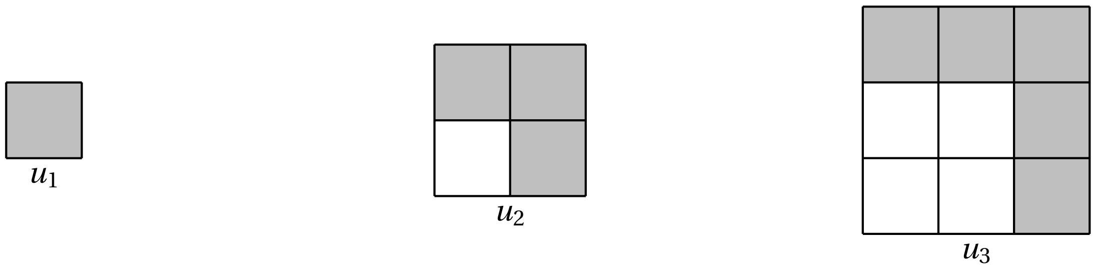
Capacité 4 ***Calculer des termes d'une suite définie explicitement, voir exo 1 p.11***

1. Calculer les trois premiers termes de la suite (v_n) définie pour tout entier $n \geqslant 1$ par $v_n = \frac{2^n + 1}{2 + (-1)^n 2^{n+1}}$.

2. Soit la suite (u_n) définie par $u_0 = 0$ et pour tout entier $n \geqslant 1$, $u_n = u_{n-1} + 2n - 1$.

a. Détailler les calculs de u_1, u_2 et u_3, puis calculer les vingt premiers termes de la suite avec la calculatrice. Quelle conjecture peut-on faire sur une formule explicite du terme général u_n ?

b. Commenter la figure ci-dessous.



2.4 Suite définie par un algorithme et listes en Python

2.4.1 Listes en Python

Méthode *Listes en* Python

- En Python, il existe un type de données structurées nommé list qui permet de stocker des collections ordonnées d'éléments. Il s'agit de l'ordre d'apparition dans la liste pas de l'ordre par comparaison d'éléments. Une liste est délimitée par les crochets [et] et ses éléments sont séparés par une virgule. Le nombre d'éléments que contient une liste est sa longueur et s'obtient avec la fonction len.

  On donne ci-dessous l'exemple de l'affectation d'une liste de notes à la variable notes.

```
>>> notes = [10, 8, 6, 11, 7]
>>> type(notes)
<class 'list'>
>>> len(notes)
5
```

- Une liste peut être définie de plusieurs façons :

  – Avec le **constructeur** list, par exemple list(range(1, 10) va générer la liste des entiers entre 1 et 9 inclus.

```
>>> L = list(range(1, 10))
>>> L
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

  – Par **extension**, en délimitant les éléments de la liste par [et] :

```
>>> L = [601, 734, 504]
>>> L
[601, 734, 504]
```

- On accède à un élément d'une liste par son index, les éléments étant indexés de gauche à droite à partir de 0 pour le premier élément. Les éléments de la liste peuvent ainsi être lus ou modifiés.

```
>>> notes[0], notes[len(notes)-1]
10, 7
>>> notes[0] = 9
>>> notes
[9, 8, 6, 11, 7]
```

- On peut ajouter un élément à la fin d'une liste avec la fonction append. On peut ainsi peupler une liste vide notée []. La réciproque de la fonction append est la fonction pop, utilisée sans argument elle extrait le dernier élément de la liste et le retourne. Avec un index en argument, elle extrait et retourne l'élément à cet index. append et pop s'utilisent avec la notation pointée car il s'agit de fonctions spécifique aux objets de type list.

```
>>> notes = []
>>> notes.append(8)
>>> notes
[8]
>>> notes.append(14)
>>> notes
[8, 14]
>>> notes.pop()
14
>>> notes
[8]
```

- Un parcours de liste peut s'effectuer de deux façons avec une boucle `for` : en parcourant les index ou directement les valeurs des éléments.

Parcours sur les index

```
notes = [10, 8, 6, 11, 7]
for k in range(len(notes)):
    #affichage des notes
    v = notes[k]
    print(v)
```

Parcours sur les valeurs

```
notes = [10, 8, 6, 11, 7]
for v in notes:
    #affichage des notes
    print(v)
```

- On peut définir directement une liste définie par une formule appliquée à tous les éléments d'une autre liste ou d'un intervalle d'entiers défini par `range`. On parle de **définition par compréhension**.

  - Liste des carrés des entiers entre 1 et 9 :

```
>>> L1 = [n ** 2 for n in range(1, 10)]
>>> L1
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

  - Liste des doubles des carrés des entiers entre 1 et 9 :

```
>>> L2 = [2 * n for n in L1]
>>> L2
[2, 8, 18, 32, 50, 72, 98, 128, 162]
```

  - Liste des entiers compris entre -4 et 4 dont le carré est supérieur à 3 :

```
>>> L3 = [n for n in range(-4, 5) if n ** 2 > 3]
>>> L3
[-4, -3, -2, 2, 3, 4]
```

Capacité 5 ***Manipuler les listes en*** `Python`

Pour chaque question, sélectionner la ou les bonne(s) réponse(s).

1. On définit la liste L par `L = [852, 843, 954]`

a. `L[1]` vaut 852 **b.** `L[0]` vaut 852 **c.** `L[3]` n'est pas défini

2. On définit la liste `L` par `L = [k * 2 - 1 for k in range(3)]`

   a. `L` vaut `[1, 3, 5]` **b.** `L` vaut `[-1,1,3]` **c.** `L` vaut `[-1, 0, 3]`

3. Soit un programme `Python` :

```
from math import sin
L = []
for k in range(1, 50):
    if sin(k) >= 0:
        L.append(k)
```

   La valeur de la liste `L` à la fin du programme peut être générée par :

   a. `[sin(k) for k in range(1,50) if sin(k) >= 0]`
   b. `[k for k in range(1,50) if sin(k) >= 0]`
   c. `[k if sin(k) >= 0 for k in range(1,50)]`
   d. `[k for k in [i for in range(1, 50) if sin(i) >= 0]]`

4. Soit un programme `Python` :

```
L = list(range(2, 5))
L.pop()
L.append(14)
L.pop(1)
L.pop(1)
L.append(16)
```

   La valeur de la liste `L` à la fin du programme est :

   a. `[5,14,16]` **b.** `[2,16]` **c.** `[14,16]`

2.4.2 Suite définie par un algorithme

Algorithmique 1 *Suite de Syracuse, voir TP3 p. 25*

A la fin des années 20 du vingtième siècle, Lothar Collatz, mathématicien allemand (1910 - 1990), s'intéressait aux itérations de fonctions prenant des valeurs entières et inventa une suite de nombres (devenue célèbre à l'université de Syracuse aux USA) définie par l'algorithme suivant :

- on choisit un entier naturel;
- si cet entier est pair, on le divise par deux et sinon on le multiplie par 3 et on ajoute 1;
- on répète le procédé avec l'entier obtenu ...

1. Calculer les dix premiers termes de la suite en choisissant comme premier terme un entier au hasard entre 10 et 20. Comparer avec les voisins. Quelle conjecture peut-on faire?

   À ce jour, on a vérifié cette conjecture pour tout entier inférieur à $20 \times 2^{58} \approx 5,764 \times 10^{18}$.

2. Recopier et compléter la fonction Python ci-dessous pour qu'elle retourne la liste des n premiers termes de cette suite si on choisit pour premier terme u :

```
def syracuse(u , n):
    L = [u]
    for k in range(n):
        if ................:
            u = ...........
        else:
            u = ...........
        ..................... #ajout de u à la fin de la liste L
    return L
```

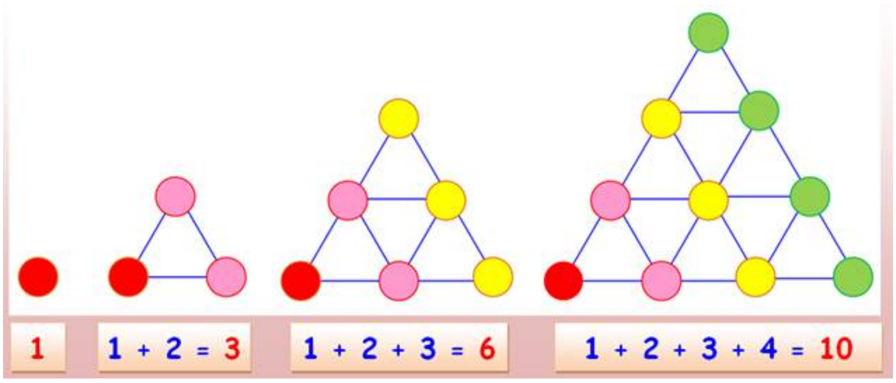
3. Écrire une fonction tempsVol(u) qui retourne le plus petit indice du terme de la suite égal à 1, si on prend comme premier terme u.

```
In [4]: tempsVol(634)
Out[4]: 38
```

2.5 Suite définie par des motifs géométriques ou combinatoires

Capacité 6 *Déterminer une relation explicite ou une relation de récurrence pour une suite définie par un motif géométrique*

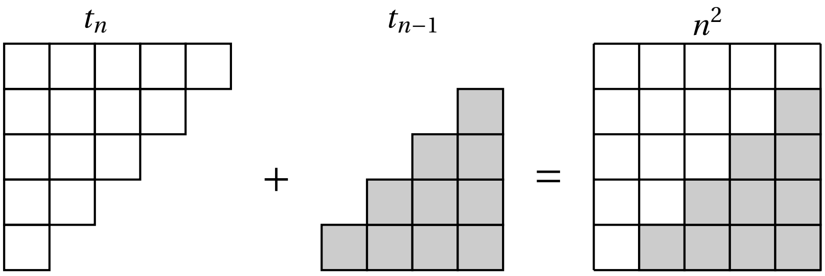
Les mathématiciens grecs représentaient certains nombres géométriquement, comme par exemple les nombres triangulaires. Si on note t_n le n^e nombre triangulaire, on a $t_1 = 1$, $t_2 = 1+2 = 3$, $t_3 = 1+2+3 = 6$, $t_4 = 1+2+3+4 = 10\ldots$



Source : http://www.villemin.gerard.free.fr

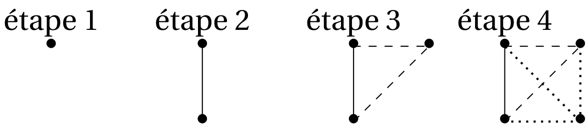
1. Établir une relation de récurrence entre t_n et t_{n-1} pour tout entier $n \geqslant 2$.

2. On admet que pour tout entier $n \geqslant 2$, $t_n + t_{n-1} = n^2$. On donne ci-dessous une preuve géométrique.



En déduire une formule explicite de t_n pour tout entier $n \geqslant 2$.

3. On considère la suite de motifs ci-dessous dans laquelle on rajoute un point à chaque étape. Pour tout entier naturel $n \geqslant 1$, on note u_n le nombre de segments tracés sur le n^{e} motif.



a. Exprimer pour tout entier $n \geqslant 1$, u_{n+1} en fonction de u_n.

b. Déterminer une formule explicite donnant u_n en fonction de n pour tout entier $n \geqslant 1$.

4. Dans une assemblée de n personnes, où chacune salue exactement une fois toutes les autres, 45 poignées de main sont échangées. Combien de personnes compte cette assemblée ?

Algorithmique 2 *Déterminer une relation explicite ou une relation de récurrence pour une suite définie par une question de dénombrement*

1. Trois chevaux d'égale valeur sont au départ d'une course. Déterminer avec un arbre de dénombrement le nombre de classements possibles à l'arrivée. Comment déterminer le nombre de classements pour une course de 6 chevaux ?

2. La **factorielle** d'un entier positif n est le terme général de la suite (u_n) définie par :

$$u_n = \begin{cases} 0 \text{ si } n = 0 \\ 1 \times 2 \times 3 \times \ldots \times (n-1) \times n \text{ si } n \geqslant 1 \end{cases}$$

Pour tout entier $n \geqslant 0$, on note $u_n = n!$

a. Exprimer u_{n+1} en fonction de u_n pour tout entier $n \geqslant 0$.

b. Écrire en `Python` une fonction `factorielle(n)` qui retourne $u_n = n!$ pour tout entier $n \geqslant 0$.

3 Évolution linéaire ou évolution exponentielle

Histoire 2

Dans ses *Recherches générales sur la mortalité et la multiplication du genre humain*, Leonhard Euler (1707 – 1783), modélise l'évolution de la population annuelle d'une ville par la relation de récurrence $p_{n+1} = \lambda p_n$ avec λ le rapport des naissances de l'année $n+1$ sur les naissance de l'année n supposé indépendant de n. Il s'agit d'un modèle d'évolution exponentiel.

Thomas Malthus (1766 – 1834) reprend l'idée d'une évolution exponentielle de la population (suite géométrique) et fait l'hypothèse que la capacité production suit plutôt une évolution linéaire (suite arithmétique) ce qui l'amène à préconiser une limitation des naissances.

Ce modèle est remis en cause vers 1840 par Pierre François Verhulst (1804 – 1849), dont le modèle logistique prend en compte la limitation de la population : la population p_{n+1} à l'année $n+1$ est proportionnelle à $p_n(M - p_n)$ où M est la population maximale contrainte par des facteurs limitants (place, nourriture ...).

Capacité 7 ***Modéliser un phénomène discret à croissance linéaire***

Un vendeur de logiciel propose un contrat d'assistance de deux ans maximum, comprenant l'installation à domicile et un conseiller joignable par téléphone pour 20 € le premier mois, puis 0,6 € de moins par rapport au mois précédent, et ainsi de suite. On note v_n la mensualité en euros au $n^{\text{ième}}$ mois.

1. Calculer v_1 et v_2.
2. Pour tout entier n tel que $1 \leqslant n \leqslant 23$, exprimer v_{n+1} en fonction de v_n.
3. Calculer le montant de la mensualité au $12^{\text{ième}}$ mois et déterminer sans justifier une formule explicite de v_n pour tout entier n tel que $1 \leqslant n \leqslant 24$.

Capacité 8 ***Modéliser un phénomène discret à croissance exponentielle, voir la situation C page 9***

Au premier Janvier 2 016, on place un capital de 800 euros sur un compte bloqué, rémunéré au taux d'intérêts composés annuels de 2,5 %. On note C_n le capital en $2\,016 + n$.

1. Calculer C_1 et C_2, arrondir au centime près.
2. Pour tout entier naturel n, exprimer C_{n+1} en fonction de C_n puis déterminer sans justifier une formule explicite de C_n.
3. Recopier et compléter la fonction ci-dessous et sa traduction en `Python` pour qu'elles retournent la première année où le capital dépassera `s` euros.
4. Faire un tableau des valeurs de C_n avec le mode Suite de la calculatrice pour déterminer la valeur de `seuil(1000)`.

Algorithme

```
Fonction seuil(s):
  c ⟵ 800
  n ⟵ 0
  Tant que ......
    c ⟵ ...
    n ⟵ n + 1
  Retourne n
```

Python

```python
def seuil(s):
    c = 800
    n = 0
    while c ... s:
        c = .........
        n = n + 1
    return n
```

4 Suites arithmétiques

4.1 Définition

Définition 5

Une suite (u_n) est **arithmétique** s'il existe un réel r tel que :

$$\text{pour tout entier } n \in \mathbb{N}, \quad u_{n+1} = u_n + r$$

$$u_n \xrightarrow{+r} u_{n+1}$$

Le réel r est la **raison** de la suite.

Propriété 1 *Suite arithmétique et variation absolue*

Une suite $(u_n)_{n\in\mathbb{N}}$ est **arithmétique** si la **variation absolue** $u_{n+1} - u_n$ entre deux termes consécutifs quelconques est égale à une constante r.

Capacité 9 *Caractériser une suite arithmétique*

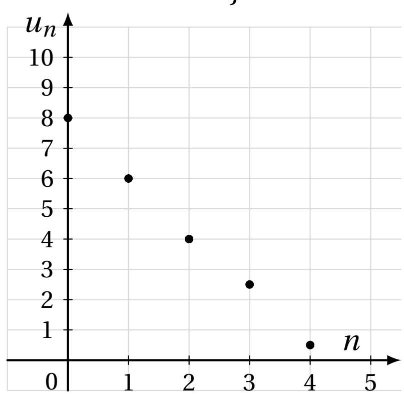
Pour chacune des affirmations suivantes, déterminer si elle est *Vraie* ou *Fausse* en justifiant la réponse.

- **Affirmation 1 :**

  La suite (u_n) dont on a représenté ci-contre les termes u_0, u_1, u_2 et u_3 peut être arithmétique.

- **Affirmation 2 :**

  La suite $(v_n)_{n\geqslant 0}$ de terme général $v_n = (n+1)^2 - (n-2)^2$, est arithmétique.



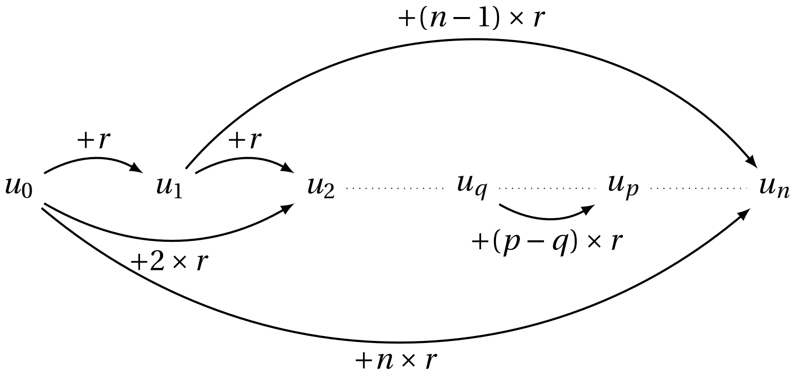
4.2 Calculs de termes et représentation graphique

Propriété 2

Soit $(u_n)_{n\in\mathbb{N}}$ une suite arithmétique de raison r.

- Pour tout entier $n \in \mathbb{N}$, $u_n = u_0 + \cdots\cdots \times r$ et $u_n = u_1 + \cdots\cdots \times r$
- Pour tous entiers $p \in \mathbb{N}$ et $q \in \mathbb{N}$, $u_p = u_q + \cdots\cdots \times r$.

Une suite arithmétique est entièrement déterminée par la connaissance d'un terme et de sa raison.



Démonstration ***détaillée p. 20***

Capacité 10 ***Calculer le terme général d'une suite arithmétique, voir exos 1 et 2 p. 13***

1. Soit (u_n) une suite arithmétique de raison -3 et de premier terme $u_1 = 10$.

a. Exprimer u_{n+1} en fonction de u_n.

b. On utilise un tableur pour calculer les termes de la suite (u_n). Proposer une formule à écrire en B3 pour calculer u_2 ; cette formule « tirée vers le bas » dans la colonne devra permettre de calculer les valeurs successives de la suite (u_n).

	A	B
1	n	u_n
2	1	10
3	2	

c. Exprimer u_n en fonction de n et calculer u_{43}.

2. Soit (w_n) une suite arithmétique telle que $w_{13} = -5$ et $w_{17} = 23$.

a. Soit r la raison de cette suite, exprimer $w_{17} - w_{13}$ en fonction de r. En déduire la valeur de r puis calculer w_1.

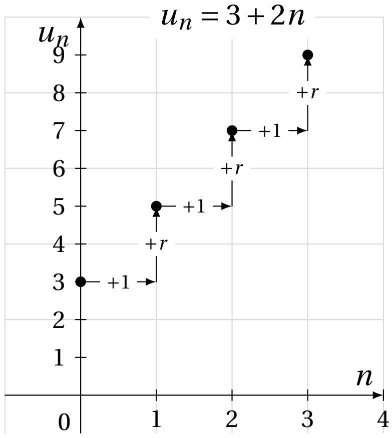
b. Exprimer w_n en fonction de n et calculer w_{50}.

Propriété 3

Une suite $(u_n)_{n\in\mathbb{N}}$ est arithmétique de raison r si et seulement s'il existe deux réels a et r tels que :

$$\text{pour tout entier } n \in \mathbb{N}, \quad u_n = a + nr$$

Une suite est arithmétique si et seulement si sa représentation graphique dans un repère est constituée de points alignés.
Une suite arithmétique modélise une **évolution linéaire**.



Démonstration

- $\boxed{\Longrightarrow}$ Soit une suite arithmétique $(u_n)_{n\geqslant 0}$ de raison r.
  D'après la propriété précédente, pour tout entier $n \geqslant 0$, on a $u_n = u_0 + nr$ donc $u_n = a + nr$ avec $a = u_0$.
- $\boxed{\Longleftarrow}$ Soit une suite arithmétique $(u_n)_{n\geqslant 0}$ telle pour tout entier $n \geqslant 0$, on a $u_n = a + nr$.
  Pour tout entier $n \geqslant 0$, on a $u_{n+1} - u_n = a + (n+1)r - (a + nr) = r$ donc $(u_n)_{n\geqslant 0}$ est arithmétique de raison r.

4.3 Somme des termes successifs

Propriété 4

Soit (u_n) une suite arithmétique de raison r.
Pour tout $n \in \mathbb{N}$ on a :

$$\sum_{k=0}^{n} u_k = u_0 + u_1 + \cdots + u_{n-1} + u_n = (n+1) \times \frac{u_0 + u_n}{2}$$

Plus généralement, pour tout $p \in \mathbb{N}$, pour tout $q \in \mathbb{N}$ tel que $q \geqslant p$ on a :

$$\sum_{k=p}^{q} u_k = u_p + u_{p+1} + \cdots + u_{q-1} + u_q = (q-p+1) \times \frac{u_p + u_q}{2} = \text{nombre de termes} \times \frac{\text{premier terme} + \text{dernier terme}}{2}$$

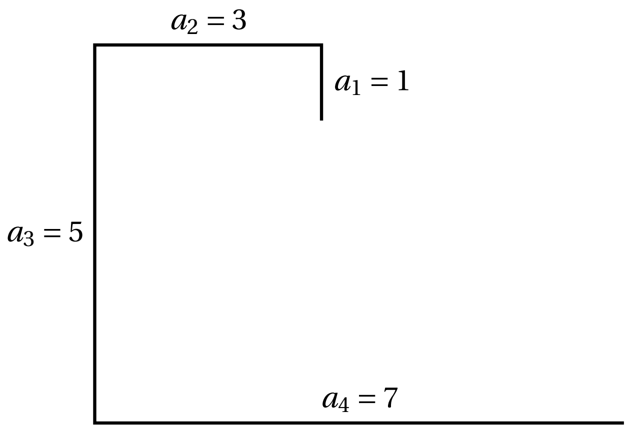
En particulier pour la suite arithmétique telle que pour tout $n \in \mathbb{N}$, $u_n = n$ on a :

$$\sum_{k=1}^{n} k = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

Démonstration *guidée p. 21*

Capacité 11 *Déterminer une relation pour une suite définie par un motif géométrique, voir exo 3 p. 13*

Un robot d'exploration spatiale se déplace en spirale sur la planète qu'il doit explorer. Il avance d'abord de 1 mètre puis tourne à gauche à angle droit, il avance de trois mètres puis tourne à gauche à angle droit et ainsi de suite. A chaque étape il incrémente de deux mètres son déplacement rectiligne.
On note a_n la longueur du $n^{\text{ième}}$ déplacement rectiligne du robot.



1. Déterminer la nature de la suite (a_n) puis une formule explicite de son terme général a_n.
2. Déterminer une formule explicite de la distance totale parcourue par le robot après n déplacements en ligne droite.
3. Le robot avance à la vitesse de 20 centimètres par seconde en ligne droite et met deux secondes

pour tourner. Calculer le temps nécessaire pour que le robot parcourt exactement 1 kilomètre.

5 Suites géométriques

5.1 Définition

Définition 6

Une suite $(u_n)_{n\in\mathbb{N}}$ est **géométrique** s'il existe un réel q tel que :

$$\text{pour tout entier } \in \mathbb{N}, \quad u_{n+1} = q \times u_n$$

$$u_n \xrightarrow{\times q} u_{n+1}$$

Le réel q est la **raison** de la suite.

Propriété 5

Une suite $(u_n)_{n\in\mathbb{N}}$ qui ne s'annule jamais est **géométrique** si et seulement si la **variation relative** $\dfrac{u_{n+1}-u_n}{u_n}$ entre deux termes consécutifs quelconques est égale à une constante, c'est-à-dire si le **taux d'évolution** entre deux termes consécutifs est constant.
Une suite géométrique modélise une **évolution exponentielle**.

Démonstration

Soit une suite $(u_n)_{n\in\mathbb{N}}$ qui ne s'annule jamais.
$(u_n)_{n\in\mathbb{N}}$ est géométrique de raison $q \neq 0$ si et seulement si $\forall n \in \mathbb{N},\ u_{n+1} = qu_n$.

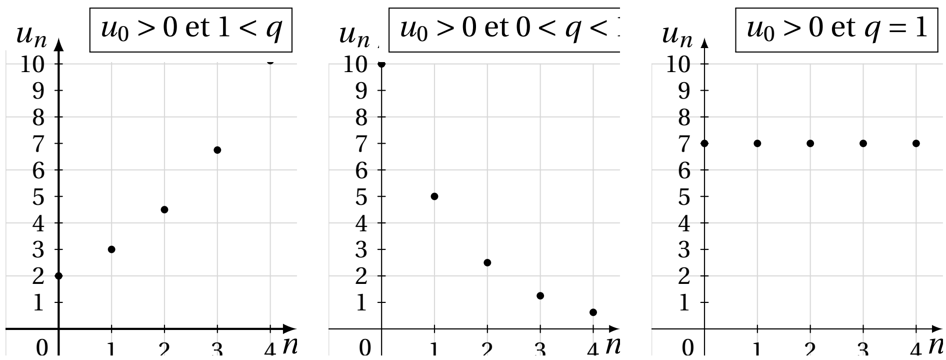
$$\forall n \in \mathbb{N},\ u_{n+1} = qu_n \Longleftrightarrow \forall n \in \mathbb{N},\ u_{n+1} - u_n = (q-1)u_n$$

on peut diviser par $u_n \neq 0$

$$\forall n \in \mathbb{N},\ u_{n+1} = qu_n \Longleftrightarrow \forall n \in \mathbb{N},\ \frac{u_{n+1}-u_n}{u_n} = q-1$$

Propriété 6 ***Représentation graphique d'une suite géométrique***

Dans un repère du plan, la représentation graphique d'une suite géométrique $(u_n)_{n\in\mathbb{N}}$ de raison q n'est pas constituée de points alignés sauf dans le cas où $q = 1$.

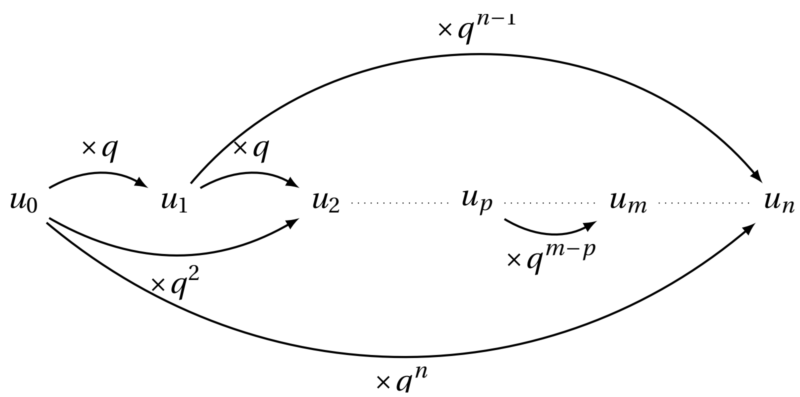


5.2 Calculs de termes

Propriété 7

Soit $(u_n)_{n\in\mathbb{N}}$ une suite géométrique de raison q.

- Pour tout entier $n \in \mathbb{N}$, $u_n = u_0 \times q^n$ et $u_n = u_1 \times q^{n-1}$
- Pour tous entiers $p \in \mathbb{N}$ et $m \in \mathbb{N}$, $u_m = u_p \times q^{m-p}$.



Démonstration ***détaillée p. 20***

Capacité 12 ***Modéliser un phénomène discret à croissance exponentielle, calculer le terme général d'une suite géométrique, voir exos 1 et 2 p. 15***

La scintigraphie cardiaque est une technique d'imagerie qui permet d'examiner la qualité de l'irrigation du cœur par les artères coronaires.
Lors de cet examen, on injecte au patient un échantillon d'un isotope de Thallium d'activité radioactive 60 MBq (Méga Becquerel).
On appelle demi-vie le temps mis par une substance radioactive pour perdre la moitié de son activité.
Ainsi, après une demi-vie, l'activité radioactive de cet échantillon de Thallium est de 30 MBq et après

deux demi-vies, l'activité radioactive de cet échantillon est de 15 MBq.
On note u_0 l'activité radioactive de cet échantillon (en MBq) à l'injection et u_n l'activité radioactive de cet échantillon (en MBq) après n demi-vies avec n entier naturel.

1. Donner les valeurs de u_0, u_1, u_2 et u_3.

2. Exprimer u_{n+1} en fonction de u_n. En déduire la nature de la suite (u_n).

3. **a.** Exprimer u_n en fonction de n.

b. Déterminer l'activité radioactive de cet échantillon après 5 demi-vies.

4. Écrire en `Python` une fonction `seuil()` qui retourne le plus petit entier n à partir duquel $u_n < 0,25$.

5. Sachant que la demi-vie de cet isotope de Thallium est d'environ 3 jours, déterminer le nombre de jours au bout desquels on est certain que l'activité radioactive de cet échantillon est strictement inférieure à 0,25 MBq.

5.3 Somme des termes successifs

Propriété 8

Soit (u_n) une suite géométrique de raison $q \neq 1$.
Pour tout $n \in \mathbb{N}$ on a :

$$\sum_{k=0}^{n} u_k = u_0 + u_1 + \cdots + u_{n-1} + u_n = u_0 \times \frac{1-q^{n+1}}{1-q}$$

Plus généralement, Pour tout $p \in \mathbb{N}$, pour tout $m \in \mathbb{N}$ tel que $m \geqslant p$ on a :

$$\sum_{k=p}^{m} u_k = u_p + u_{p+1} + \cdots + u_{m-1} + u_m = u_p \times \frac{1-q^{m-p+1}}{1-q} = \text{premier terme} \times \frac{1-raison^{\text{nombre de termes}}}{1-raison}$$

En particulier pour la suite géométrique telle que pour tout $n \in \mathbb{N}$, $u_n = q^n$ avec $q \neq 1$ on a :

$$\sum_{k=0}^{n} q^k = 1 + q + q^2 + \cdots + q^n = \frac{1-q^{n+1}}{1-q}$$

Démonstration ***guidée p. 21***

☞ On démontre d'abord que pour tout réel $q \neq 1$ on a $1 + q + q^2 + \cdots + q^n = \dfrac{1-q^{n+1}}{1-q}$:

....

....

....

....

....

☞ On démontre ensuite la formule de la somme des termes consécutifs d'une suite géométrique (u_n) de raison $q \neq 1$:

....

....

....

....

Capacité 13 ***Calculer la somme de termes consécutifs d'une suite géométrique, voir exo 3 p.15***

Pour tout entier naturel n, on considère la somme $s_n = 1 + \frac{1}{2} + \ldots + \frac{1}{2^n} = \sum_{k=0}^{n} \frac{1}{2^k}$.

1. Exprimer s_n en fonction de n.
2. Quelle conjecture peut-on formuler sur la valeur de s_n lorsque n tend vers $+\infty$?

Capacité 14 ***Déterminer une relation explicite ou une relation de récurrence pour une suite définie par un motif géométrique ou une question de dénombrement***

Une institutrice propose un atelier découpage pour ses élèves à partir d'une feuille de 400 cm^2.

Étape 1 l'élève partage d'abord la feuille en 9 carrés et découpe le carré central ;

Étape 2 l'élève partage alors les 8 carrés restant en 9 carrés égaux et découpe le carré central ;

Étapes suivantes l'élève répète le même procédé ...

1. On note u_n la surface restante de la feuille après n découpes. Ainsi $u_0 = 400$.
   a. Justifier que la suite (u_n) est géométrique et déterminer sa raison.
   b. Que peut-on conjecturer pour les valeurs de u_n lorsque n devient aussi grand que l'on veut ?
   c. Recopier et compléter la fonction `seuil(s)` pour qu'elle retourne le plus petit entier n tel que $u_n \leqslant$ `s`.
   Programmer cette fonction, quelle est la valeur retournée par `seuil(10)` ?
2. On note v_n le nombre de nouveaux carrés découpés lors de la $n^{ième}$ découpe avec $n \geqslant 1$. Ainsi $v_1 = 1$, $v_2 = 8$ ...

a. Justifier que la suite (v_n) est géométrique et déterminer sa raison.

b. Que peut-on conjecturer pour les valeurs de v_n lorsque n devient aussi grand que l'on veut ?

c. Recopier et compléter la fonction `somme(n)` pour qu'elle retourne t_n, le nombre total de carrés découpés après n découpes avec $n \geqslant 1$.
Programmer cette fonction, quelle est la valeur retournée par `somme(10)` ?

d. Déterminer une formule explicite permettant de calculer t_n pour un entier $n \geqslant 1$.

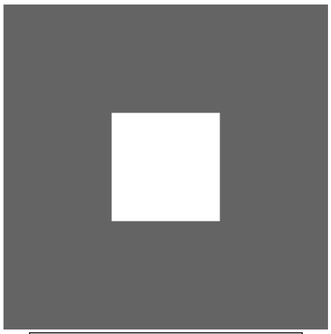
Algorithme de seuil

```
Fonction seuil(s):
  n ⟵ 0
  u ⟵ 400
  Tant que ......
    u ⟵ ...
    n ⟵ n + 1
  Retourne n
```

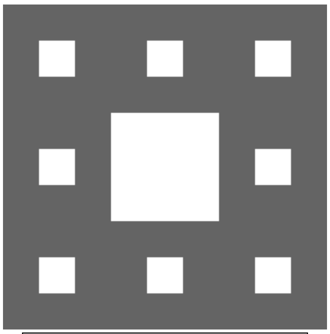
Python

```python
def seuil(s):
    n = 0
    u = 400
    while ..... :
        u = .........
        n = n + 1
    return n
```

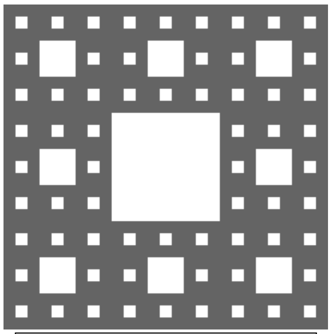
Étape 0 : 0 découpe



Étape 1 : 1 découpe



Étape 2 : 9 découpes



Étape 3 : 73 découpes

Calcul de somme

```
Fonction somme(n):
  v ⟵ 1
  t ⟵ ...
  Pour k allant de 2 à n
    v ⟵ ...
    t ⟵ ...
  Retourne t
```

Python

```python
def somme(n):
    v = 1
    t = ........
    for k in range(2, n + 1):
        v = .........
        t = .........
    return t
```