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 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* :

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

1.2 Modèle d'évolution d'un phénomène discret

Activité 1 *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 1er 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

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

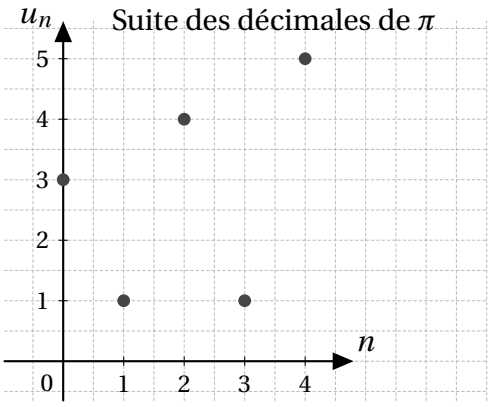
b. Quelle est la valeur retournée par l'appel de fonction `seuil()`, répondre en complétant un tableau d'évolution des variables `U` et `N`.

2 Différents modes de génération d'une suite

2.1 Suite définie par extension

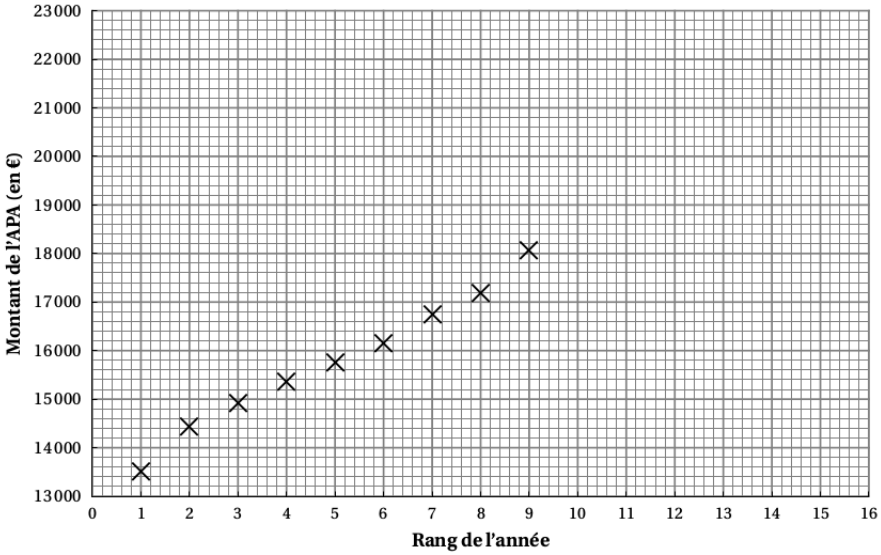
Définition 2

- 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 $(f_n)_{n\in\mathbb{N}}$ 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, $u_{n+1} = -\dfrac{1}{2}u_n + 2$.
   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}.

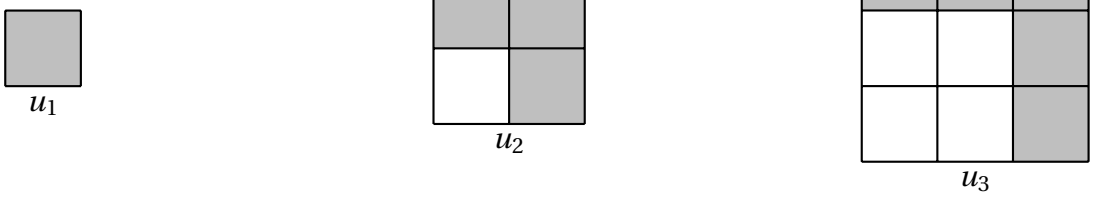
2.3 Suite définie par une formule explicite

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 $v_n = \dfrac{2^n + 1}{2 + (-1)^n 2^{n+1}}$.
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 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]`

Capacité 6 *Modéliser une situation par une suite*

On s'intéresse à une population de sangliers dans une forêt domaniale.
Au premier septembre 2021, la population était de $p_0 = 50$ individus.
Chaque année un prélèvement de 10 individus est effectué pendant la période de chasse entre le premier septembre et le premier mars de l'année suivante.
Les recensements effectués les années précédentes ont permis de conjecturer que la population de sangliers augmente de 10 % en dehors de la période de chasse, entre le premier mars et le premier septembre de chaque année.
On considère que la population de sangliers va suivre ce modèle dans les prochaines années et on note p_n la population de sangliers au premier septembre 2021 + n. Ainsi on a $p_0 = 50$.
La suite (p_n) donnant l'évolution de la population de sangliers en fonction du temps, constitue une *série chronologique*.

1. Justifier que $p_1 = 44$.

2. Soit n un entier naturel, exprimer p_{n+1} en fonction de p_n.

3. On admet désormais que pour tout entier $n \geqslant 0$, on a $p_n = 110 - 1,1^n \times 60$.

a. Compléter la fonction Python ci-dessous pour qu'elle renvoie la liste des m premiers termes de la suite de p_0 à p_{m-1}.

```
def obelix(m):
    lis = [50]
    for n in range(1, m):
        lis.append(.....................)
    return lis
```

b. On évalue `res = obelix(8)` où `res` est la liste renvoyée par `obelix(8)`.
Quelle est la valeur du dernier élément de la liste, `res[7]` ? Comment peut-on interpréter ce résultat ?

2.5 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
```

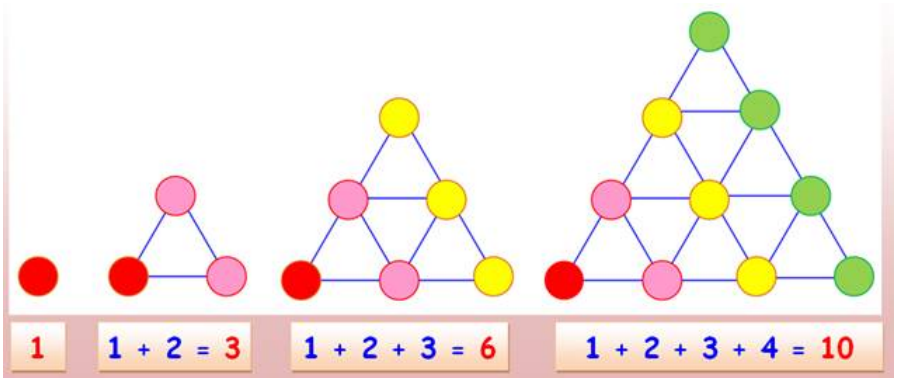
3. Écrire une fonction `tempsVol(u)` qui retourne le plus petit indice du terme de la suite égal à 1, si on prend comme premier terme `u`.

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

2.6 Suite définie par des motifs géométriques ou combinatoires

Capacité 7 *Déterminer une relation explicite ou une relation de récurrence pour une suite définie par un motif géométrique*

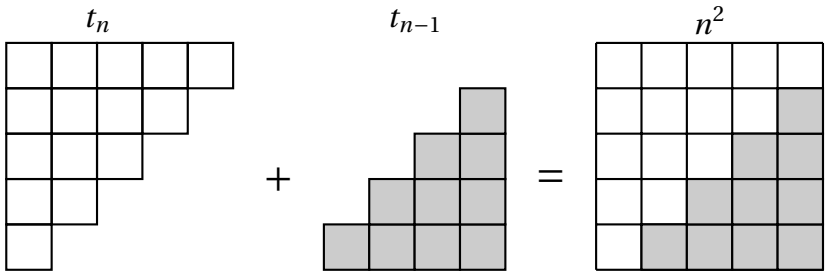
Les mathématiciens grecs représentaient certains nombres géométriquement, comme par exemple les nombres triangulaires. Si on note t_n le n^{e} nombre triangulaire, on a $t_1 = 1$, $t_2 = 1+2 = 3$, $t_3 = 1+2+3 = 6$, $t_4 = 1+2+3+4 = 10\ldots$



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

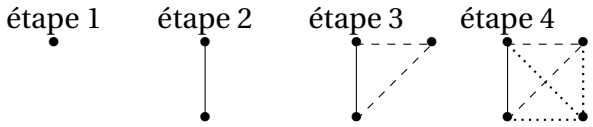
1. Établir une relation de récurrence entre t_n et t_{n-1} pour tout entier $n \geqslant 2$.

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 ?