CORRECTION DU DEVOIR SURVEILLÉ N°5- SUJET A

EXERCICE 2 : Tangente - Lecture graphique **2,5 points**

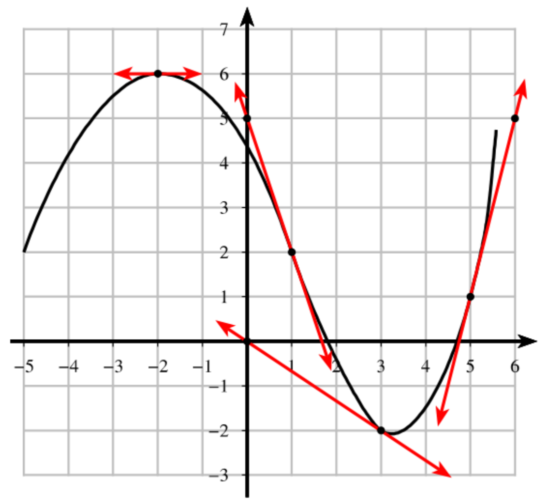
A l'aide de la représentation graphique ci-contre de la fonction f, donner les valeurs suivantes :

$f(-2) = 6$ $\quad f'(-2) = 0$

$f(1) = 2$ $\quad f'(1) = -3$

$f(3) = -2$ $\quad f'(3) = -\dfrac{2}{3}$

$f(5) = 1$ $\quad f'(5) = 4$



EXERCICE 3 : Calcul de dérivées **3,5 points**

Calculer les dérivées des fonctions suivantes sans se préoccuper des ensembles de dérivation.

1. $f(x) = -6x^3 + 5x^2 - 10x - 4$

   $f'(x) = -6 \times 3x^2 + 5 \times 2x - 10 \times 1 + 0$
   $f'(x) = -18x^2 + 10x - 10$

2. $g(x) = \dfrac{2}{x} - 3\sqrt{x}$

   $g'(x) = 2 \times \dfrac{-1}{x^2} - 3 \times \dfrac{1}{2\sqrt{x}}$

   $g'(x) = -\dfrac{2}{x^2} - \dfrac{3}{2\sqrt{x}}$

3. $h(x) = \dfrac{4x-5}{x^2}$

   h est de la forme $\dfrac{u}{v}$ avec $\begin{cases} u(x) = 4x - 5 & v(x) = x^2 \\ u'(x) = 4 & v'(x) = 2x \end{cases}$

$h' = \dfrac{u'v - uv'}{v^2}$
$h'(x) = \dfrac{4 \times x^2 - (4x-5) \times 2x}{(x^2)^2}$
$h'(x) = \dfrac{4x^2 - 8x^2 + 10x}{x^4}$
$h'(x) = \dfrac{-4x^2 + 10x}{x^4}$
$h'(x) = \dfrac{-4x + 10}{x^3}$ (on a simplifié par x car c'est un facteur commun du numérateur et du dénominateur).

Exercice 4 : Suites — 5 points

1. Soit la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_n = \dfrac{n}{n+1}$.

   (a) Étudier le sens de variation de cette suite.

   Tout d'abord on calcule le terme suivant :
   $u_{n+1} = \dfrac{n+1}{n+1+1} = \dfrac{n+1}{n+2}$
   On calcule ensuite la différence entre deux termes consécutifs :
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} - \dfrac{n}{n+1}$
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} \times \dfrac{n+2}{n+2} - \dfrac{n}{n+1} \times \dfrac{n+1}{n+1}$
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} \times \dfrac{n+1}{n+1} - \dfrac{n}{n+1} \times \dfrac{n+2}{n+2}$
   $u_{n+1} - u_n = \dfrac{(n+1)^2 - n(n+1)}{(n+1)(n+2)}$
   $u_{n+1} - u_n = \dfrac{n^2 + 2n + 1 - n^2 - n}{(n+1)(n+2)}$
   $u_{n+1} - u_n = \dfrac{1}{(n+1)(n+2)}$
   Comme n est un entier, $n+1$ et $n+2$ le sont aussi et sont positifs, donc la différence $u_{n+1} - u_n$ est toujours positive.
   Autrement dit :
   $u_{n+1} - u_n \geq 0$ ce qui est équivalent à $u_{n+1} \geq u_n$.
   La suite $(u_n)_{n\in\mathbb{N}}$ est croissante.

   (b) On souhaite programmer cette suite en Python. Compléter le programme ci-dessous :

```python
def suite(n):
    return n/(n+1)
```

2. Soit la suite $(v_n)_{n\in\mathbb{N}}$ définie par $\begin{cases} v_0 = -1 \\ v_{n+1} = 2v_n + 3 \end{cases}$. On souhaite programmer cette suite en Python. Compléter le programme ci-dessous :

```python
def suite(n):
    # initialisation
    u = -1
    # récurrence
    for i in range(n):
        u = 2*u + 3
    return u
```

3. Soit la suite arithmétique $(w_n)_{n\in\mathbb{N}}$ de terme initiale 3 et de raison $r = 2$.

   (a) Exprimer cette suite sous sa forme récurrente puis sous sa forme explicite.

   Par récurrence : $\begin{cases} w_0 = 3 \\ w_{n+1} = w_n + 2 \end{cases}$ Formule explicite : $w_n = 3 + 2n$

   (b) On sait que $w_{2018} = 4039$. En déduire w_{2022}.

   $w_{2022} = w_{2018} + 4r = 4039 + 4 \times 2 = 4039 + 8 = 4047$

4. Soit la suite géométrique $(x_n)_{n\in\mathbb{N}}$ de terme initiale 2 et de raison $q = 3$.

   (a) Exprimer cette suite sous sa forme récurrente puis sous sa forme explicite.

   Par récurrence : $\begin{cases} x_0 = 2 \\ x_{n+1} = 3x_n \end{cases}$ Formule explicite : $x_n = 2 \times 3^n$

   (b) On sait que $x_8 = 13\ 122$. En déduire x_{10}.

   $x_{10} = x_8 \times q^2 = 13\ 122 \times 3^2 = 13\ 122 \times 9 = 118\ 098$

5. Calculer la somme suivante : $1 + 2 + 3 + 4 + ... + 1000$

$1 + 2 + 3 + 4 + ... + 1000 = \dfrac{1000 \times 1001}{2} = 500 \times 1001 = 500\ 500$

(On a utilisé la formule $1 + 2 + 3 + ... + n = \dfrac{n(n+1)}{2}$

6. Calculer la somme suivante :$1 + 10 + 100 + 1000 + ... + 1\ 000\ 000$

$1 + 10 + 100 + 1000 + ... + 1\ 000\ 000 = 1\ 111\ 111$ (évident)

Exercice 5 : **5 points**

Au choix, problème 1 sur 4 points ou problème 2 sur 5 points.

Problème 1 : (sur 4 points)
L'Émir Hifik a conservé les bougies de ses gâteaux d'anniversaire depuis son premier anniversaire jusqu'à aujourd'hui sauf celles d'une année où il était trop malade pour fêter quoi que ce soit. Il possède actuellement 1 999 bougies. Quel âge a-t-il et quel âge avait-il lorsqu'il n'a pas pu fêter son anniversaire ?

On considère que L'Émir à n années. Il aurait donc dû avoir conservé exactement $1+2+3+...+n$ bougies. On doit donc chercher la plus petite valeur de n tel que $1+2+3+...+n$ dépasse 1999. Enfin la différence $1999-(1+2+3+...+n)$ devrait nous donner l'âge auquel L'Émir n'a pas pu fêter son anniversaire.

D'après le cours, on sait que $1+2+3+...+n=\dfrac{n(n+1)}{2}$. On doit donc rechercher la plus petite valeur de n tels que

$$\frac{n(n+1)}{2} > 1999$$

Pour trouver la valeur de n on peut procéder de proche en proche à la calculatrice. On peut aussi écrire un programme en Python, qui va calculer $\dfrac{n(n+1)}{2}$ et $1999-\dfrac{n(n+1)}{2}$ pour un certain nombre de valeurs de n. Par exemple :

```python
for n in range(100):
    somme = n*(n+1)/2
    difference = 1999-somme
    print(n, somme, difference)
```

On trouve alors pour n autour de 60 :

```
60 1830.0 169.0
61 1891.0 108.0
62 1953.0 46.0
63 2016.0 -17.0
64 2080.0 -81.0
```

On en déduit que l'Emir Hifik a 63 ans et que c'est le jour de ses 17 ans qu'il a été malade.

On aurait pu écrire un programme un peu plus efficace et qui n'affiche que le résultat attendu en utilisant une boucle **tant que** :

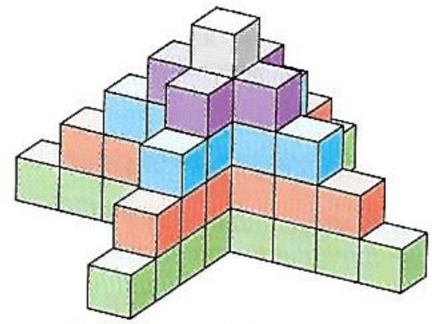
```python
n = 1
somme = 1
difference = 1998

while somme < 1999 :
    n = n + 1
    somme = n*(n+1)/2
    difference = 1999-somme
print(n, somme, difference)
```

On obtient alors uniquement le résultat attendu :

```
63 2016.0 -17.0
```

Problème 2 : (sur 5 points)

Quel est le plus grand empilement du type de celui ci-contre que l'on peut réaliser si on ne dispose que de 12 420 cubes ?



Empilement à 5 niveaux

On peut modéliser le nombre de cubes par niveaux par une suite numérique $(u_n)_{n\in\mathbb{N}}$ n représentant le niveau et u_n le nombre de cubes à ce niveau.
Au premier niveau $u_1 = 1$
Au deuxième niveau $u_2 = 5$
Au troisième niveau $u_3 = 9$
Au quatrième niveau $u_4 = 13$
Au cinquième niveau $u_5 = 17$
On constate que la suite $(u_n)_{n\in\mathbb{N}}$ est une suite arithmétique de premier terme $u_1 = 1$ et de raison $r = 4$.
On cherche donc la valeur de n la plus grande tel que la somme $u_1 + u_2 + ... + u_n$ soit inférieure à 12 420.

D'après le cours, on sait que comme la suite est arithmétique, la somme $u_1 + u_2 + ... + u_n$ s'obtient par la formule

$$\frac{(premier\ terme + dernier\ terme) \times nombre\ de\ terme}{2}$$

Le premier terme est $u_1 = 1$
Le dernier terme est $u_n = u_1 + (n-1) * r = 1 + (n-1) \times 4 = 4n - 3$
Le nombre de terme est n
Donc :

$$\frac{(premier\ terme + dernier\ terme) \times nombre\ de\ terme}{2} = \frac{(1 + 4n - 3) \times n}{2} = \frac{(4n-2) \times n}{2} = (2n-1)n$$

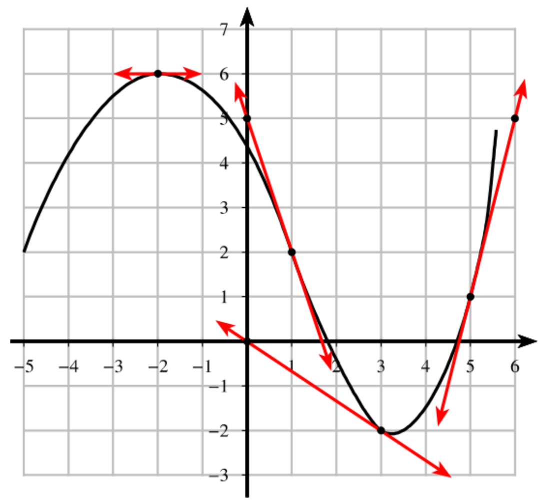
Il ne reste donc plus qu'à trouver la plus grande valeur de n tel que $(2n-1)n \leq 12\ 420$. On peut le faire de proche en proche à la calculatrice. On peut aussi le faire avec un programme en Python en utilisant une boucle **pour** et afficher tous les résultats. Mieux, on peut écrire un programme en Python utilisant une boucle **tant que** :

```python
n = 1
somme = 1

while somme < 12420 :
    n = n + 1
    somme = (2*n-1)*n
# attention, on a depasse 12420. Il faut prendre la valeur précédente

n = n - 1
somme = (2*n-1)*n

print(n, somme)
```

On obtient le résultat suivant dans la console :
`79 12403`
Donc, on peut empiler 79 niveaux de cubes avec 12 420 cubes et on en aura utiliser exactement 12 403.

Exercice 2 : Tangente - Lecture graphique **2,5 points**

A l'aide de la représentation graphique ci-contre de la fonction f, donner les valeurs suivantes :



$f(-2) = 6$ $\quad$ $f'(-2) = 0$

$f(1) = 2$ $\quad$ $f'(1) = -3$

$f(3) = -2$ $\quad$ $f'(3) = -\dfrac{2}{3}$

$f(5) = 1$ $\quad$ $f'(5) = 4$

Exercice 3 : Calcul de dérivées **3,5 points**

Calculer les dérivées des fonctions suivantes sans se préoccuper des ensembles de dérivation.

1. $f(x) = -6x^3 + 5x^2 - 10x - 4$

   $f'(x) = -6 \times 3x^2 + 5 \times 2x - 10 \times 1 + 0$
   $f'(x) = -18x^2 + 10x - 10$

2. $g(x) = \dfrac{2}{x} - 3\sqrt{x}$

   $g'(x) = 2 \times \dfrac{-1}{x^2} - 3 \times \dfrac{1}{2\sqrt{x}}$

   $g'(x) = -\dfrac{2}{x^2} - \dfrac{3}{2\sqrt{x}}$

3. $h(x) = \dfrac{4x-5}{x^2}$

   h est de la forme $\dfrac{u}{v}$ avec $\begin{cases} u(x) = 4x - 5 & v(x) = x^2 \\ u'(x) = 4 & v'(x) = 2x \end{cases}$

$h' = \dfrac{u'v - uv'}{v^2}$
$h'(x) = \dfrac{4 \times x^2 - (4x-5) \times 2x}{(x^2)^2}$
$h'(x) = \dfrac{4x^2 - 8x^2 + 10x}{x^4}$
$h'(x) = \dfrac{-4x^2 + 10x}{x^4}$
$h'(x) = \dfrac{-4x + 10}{x^3}$ (on a simplifié par x car c'est un facteur commun du numérateur et du dénominateur).

Exercice 4 : Suites — 5 points

1. Soit la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_n = \dfrac{n}{n+1}$.

   (a) Étudier le sens de variation de cette suite.

   Tout d'abord on calcule le terme suivant :
   $u_{n+1} = \dfrac{n+1}{n+1+1} = \dfrac{n+1}{n+2}$
   On calcule ensuite la différence entre deux termes consécutifs :
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} - \dfrac{n}{n+1}$
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} \times \dfrac{n+2}{n+2} - \dfrac{n}{n+1} \times \dfrac{n+1}{n+1}$
   $u_{n+1} - u_n = \dfrac{n+1}{n+2} \times \dfrac{n+1}{n+1} - \dfrac{n}{n+1} \times \dfrac{n+2}{n+2}$
   $u_{n+1} - u_n = \dfrac{(n+1)^2 - n(n+1)}{(n+1)(n+2)}$
   $u_{n+1} - u_n = \dfrac{n^2 + 2n + 1 - n^2 - n}{(n+1)(n+2)}$
   $u_{n+1} - u_n = \dfrac{1}{(n+1)(n+2)}$
   Comme n est un entier, $n+1$ et $n+2$ le sont aussi et sont positifs, donc la différence $u_{n+1} - u_n$ est toujours positive.
   Autrement dit :
   $u_{n+1} - u_n \geq 0$ ce qui est équivalent à $u_{n+1} \geq u_n$.
   La suite $(u_n)_{n\in\mathbb{N}}$ est croissante.

   (b) On souhaite programmer cette suite en Python. Compléter le programme ci-dessous :

```python
def suite(n):
    return n/(n+1)
```

2. Soit la suite $(v_n)_{n\in\mathbb{N}}$ définie par $\begin{cases} v_0 = -1 \\ v_{n+1} = 2v_n + 3 \end{cases}$. On souhaite programmer cette suite en Python. Compléter le programme ci-dessous :

```python
def suite(n):
    # initialisation
    u = -1
    # récurrence
    for i in range(n):
        u = 2*u + 3
    return u
```

3. Soit la suite arithmétique $(w_n)_{n\in\mathbb{N}}$ de terme initiale 3 et de raison $r = 2$.

   (a) Exprimer cette suite sous sa forme récurrente puis sous sa forme explicite.

   Par récurrence : $\begin{cases} w_0 = 3 \\ w_{n+1} = w_n + 2 \end{cases}$ Formule explicite : $w_n = 3 + 2n$

   (b) On sait que $w_{2018} = 4039$. En déduire w_{2022}.

   $w_{2022} = w_{2018} + 4r = 4039 + 4 \times 2 = 4039 + 8 = 4047$

4. Soit la suite géométrique $(x_n)_{n\in\mathbb{N}}$ de terme initiale 2 et de raison $q = 3$.

   (a) Exprimer cette suite sous sa forme récurrente puis sous sa forme explicite.

   Par récurrence : $\begin{cases} x_0 = 2 \\ x_{n+1} = 3x_n \end{cases}$ Formule explicite : $x_n = 2 \times 3^n$

   (b) On sait que $x_8 = 13\ 122$. En déduire x_{10}.

   $x_{10} = x_8 \times q^2 = 13\ 122 \times 3^2 = 13\ 122 \times 9 = 118\ 098$

5. Calculer la somme suivante : $1 + 2 + 3 + 4 + ... + 1000$

$1 + 2 + 3 + 4 + ... + 1000 = \dfrac{1000 \times 1001}{2} = 500 \times 1001 = 500\ 500$

(On a utilisé la formule $1 + 2 + 3 + ... + n = \dfrac{n(n+1)}{2}$

6. Calculer la somme suivante :$1 + 10 + 100 + 1000 + ... + 1\ 000\ 000$

$1 + 10 + 100 + 1000 + ... + 1\ 000\ 000 = 1\ 111\ 111$ (évident)

EXERCICE 5 : **5 points**

Au choix, problème 1 sur 4 points ou problème 2 sur 5 points.

Problème 1 : (sur 4 points)
L'Émir Hifik a conservé les bougies de ses gâteaux d'anniversaire depuis son premier anniversaire jusqu'à aujourd'hui sauf celles d'une année où il était trop malade pour fêter quoi que ce soit. Il possède actuellement 1 999 bougies. Quel âge a-t-il et quel âge avait-il lorsqu'il n'a pas pu fêter son anniversaire ?

On considère que L'Émir à n années. Il aurait donc dû avoir conservé exactement $1+2+3+...+n$ bougies. On doit donc chercher la plus petite valeur de n tel que $1+2+3+...+n$ dépasse 1999. Enfin la différence $1999-(1+2+3+...+n)$ devrait nous donner l'âge auquel L'Émir n'a pas pu fêter son anniversaire.

D'après le cours, on sait que $1+2+3+...+n = \dfrac{n(n+1)}{2}$. On doit donc rechercher la plus petite valeur de n tels que

$$\frac{n(n+1)}{2} > 1999$$

Pour trouver la valeur de n on peut procéder de proche en proche à la calculatrice. On peut aussi écrire un programme en Python, qui va calculer $\dfrac{n(n+1)}{2}$ et $1999-\dfrac{n(n+1)}{2}$ pour un certain nombre de valeurs de n. Par exemple :

```python
for n in range(100):
    somme = n*(n+1)/2
    difference = 1999-somme
    print(n, somme, difference)
```

On trouve alors pour n autour de 60 :
`60 1830.0 169.0`
`61 1891.0 108.0`
`62 1953.0 46.0`
`63 2016.0 -17.0`
`64 2080.0 -81.0`
On en déduit que l'Emir Hifik a 63 ans et que c'est le jour de ses 17 ans qu'il a été malade.

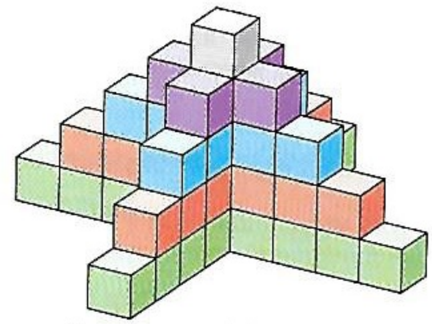
On aurait pu écrire un programme un peu plus efficace et qui n'affiche que le résultat attendu en utilisant une boucle **tant que** :

```python
n = 1
somme = 1
difference = 1998

while somme < 1999 :
    n = n + 1
    somme = n*(n+1)/2
    difference = 1999-somme
print(n, somme, difference)
```

On obtient alors uniquement le résultat attendu :
`63 2016.0 -17.0`

Problème 2 : (sur 5 points)

Quel est le plus grand empilement du type de celui ci-contre que l'on peut réaliser si on ne dispose que de 12 420 cubes ?



Empilement à 5 niveaux

On peut modéliser le nombre de cubes par niveaux par une suite numérique $(u_n)_{n\in\mathbb{N}}$ n représentant le niveau et u_n le nombre de cubes à ce niveau.
Au premier niveau $u_1 = 1$
Au deuxième niveau $u_2 = 5$
Au troisième niveau $u_3 = 9$
Au quatrième niveau $u_4 = 13$
Au cinquième niveau $u_5 = 17$
On constate que la suite $(u_n)_{n\in\mathbb{N}}$ est une suite arithmétique de premier terme $u_1 = 1$ et de raison $r = 4$.
On cherche donc la valeur de n la plus grande tel que la somme $u_1 + u_2 + ... + u_n$ soit inférieure à 12 420.

D'après le cours, on sait que comme la suite est arithmétique, la somme $u_1 + u_2 + ... + u_n$ s'obtient par la formule

$$\frac{(premier\ terme + dernier\ terme) \times nombre\ de\ terme}{2}$$

Le premier terme est $u_1 = 1$
Le dernier terme est $u_n = u_1 + (n-1) * r = 1 + (n-1) \times 4 = 4n - 3$
Le nombre de terme est n
Donc :

$$\frac{(premier\ terme + dernier\ terme) \times nombre\ de\ terme}{2} = \frac{(1 + 4n - 3) \times n}{2} = \frac{(4n-2) \times n}{2} = (2n-1)n$$

Il ne reste donc plus qu'à trouver la plus grande valeur de n tel que $(2n-1)n \leq 12\ 420$. On peut le faire de proche en proche à la calculatrice. On peut aussi le faire avec un programme en Python en utilisant une boucle **pour** et afficher tous les résultats. Mieux, on peut écrire un programme en Python utilisant une boucle **tant que** :

```python
n = 1
somme = 1

while somme < 12420 :
    n = n + 1
    somme = (2*n-1)*n
# attention, on a depasse 12420. Il faut prendre la valeur précédente

n = n - 1
somme = (2*n-1)*n

print(n, somme)
```

On obtient le résultat suivant dans la console :
`79 12403`
Donc, on peut empiler 79 niveaux de cubes avec 12 420 cubes et on en aura utiliser exactement 12 403.