Groupe de Réflexion Académique Lycée (GRAL)

en Mathématiques

Mai 2020

Gourjon – Cédric

Professeur de mathématiques

Lycée Simone Veil à Valbonne dans les Alpes Maritimes

Outils : *Sur feuille*

Nature : *bilans de connaissances et de compétences en lien avec les attendus de fin d'année (COVID)*

Objectifs pédagogiques : *Proposition d'un bilan de connaissances et de compétences utilisable par les professeurs.*

Voie : *générale*

Niveau de classe : *Première spécialité mathématiques*

Thématique(s) du programme : *Consolider ce qui a été vu en seconde et en début de première en algorithmique et programmation.*

Pré-requis : *variables, structures conditionnelles, structures itératives, notion de fonction, notion de liste.*

Voici une série d'exercices bilans conformes aux attendus de fin d'année (COVID).

Les exercices proposés ci-dessous sont de difficultés très variables. Ils permettent de vérifier les acquis de seconde (structures conditionnelles, itératives, fonctions) mais aussi ceux de première (liste).

Certains exercices comportent des algorithmes mentionnés dans le programme de première.

Les contextes sont volontairement différents (suites, géométrie analytique, variables aléatoires, statistiques de niveau seconde, dérivation) pour permettre aux professeurs d'utiliser des exercices de ce bilan même si le programme n'a pas été terminé.

Exercice I : (Vecteurs normaux , vecteurs colinéaires et orthogonaux)

On considère deux droites d_1 et d_2 de vecteurs normaux respectifs $\overrightarrow{n_1}(x_1\,;y_1)$ et $\overrightarrow{n_2}(x_2\,;y_2)$ à coordonnées entières. On souhaite programmer une fonction **posrel** *en python qui retournera la position relative des droites d_1 et d_2 : parallèles, perpendiculaires ou ni l'un, ni l'autre.*
Compléter le programme python suivant pour qu'il réponde à ce qui est demandé:

```
def posrel(x1,y1,x2,y2):
    if ................ == 0:
        return("les droites sont parallèles")
    elif ................ == 0:
        return("les droites sont perpendiculaires")
    else:
        return("les droites ne sont ni parallèles, ni perpendiculaires")
```

Et le tester pour :
a) $\overrightarrow{n_1}(-3\,;1)$ *et* $\overrightarrow{n_2}(9\,;-3)$ *b)* $\overrightarrow{n_1}(6\,;-7)$ *et* $\overrightarrow{n_2}(-8\,;7)$ *c)* $\overrightarrow{n_1}(-1;0)$ *et* $\overrightarrow{n_2}(0\,;2)$

Exercice II : (Suites, calcul de termes, de somme)

On considère la suite (v_n) définie sur $\mathbb{N}$ par $v_0 = 4$ et $v_{n+1} = v_n + \dfrac{1}{v_n}$.

1) a) Compléter le programme python suivant pour que SuiteV(n) retourne la valeur de v_n en fin d'algorithme et le programmer :

```
def SuiteV(n):
    v = 4
    for i in range(n):
        v = .......
    return(v)
```

b) A l'aide du programme, compléter le tableau suivant :

n	*0*	*1*	*2*	*3*	*4*	*5*	*6*
v_n							

2) On écrit le programme suivant à la suite du précédent :

```
def Somme(k):
    S = 0
    for n in range(k+1):
        S = S + SuiteV(n)
    return(S)
```

Quel résultat est affiché lorsqu'on exécute Somme(6) ? Justifier.

3) Exécuter Somme(100), Somme(1000) et Somme(10000). Quelle semble être la limite de la suite Somme(n) lorsque l'entier n tend vers +∞ ?

<u>Exercice III</u> : (Suites, Seuil)

On considère la suite (u_n) définie sur $\mathbb{N}$ par $u_n = 3\times2{,}5^n$.
On cherche à déterminer le rang n_1 à partir duquel les termes de la suite seront strictement supérieurs à 500.

1) Quelle est la nature de la suite (u_n) ?

2) Justifier que si n_1 existe, pour tout entier $n \geq n_1$, $u_n \geq 500$.

3) Pour déterminer n_1, on dispose de l'algorithme suivant :

```
n ← 0
u ← 3
Tant que u < 500 :
    n ← n+1
    u ← 3×2,5^n
```

Utiliser cet algorithme pour compléter le tableau suivant en ajoutant autant de lignes que nécessaire pour déterminer la valeur de n_1:

	Test $u < 500$	n	u
Etape 0		0	3
Etape 1	…	…	…

4) Programmer cet algorithme en langage python et vérifier le résultat trouvé à la question précédente.

5) a) On considère la suite (v_n) définie sur $\mathbb{N}$ par $v_n = 7 \times 0{,}2^n$.
En vous inspirant de la question 3), écrire un algorithme qui détermine le rang n_2 à partir duquel les termes de la suite (v_n) seront inférieurs à 10^{-3}. Le programmer en python et déterminer la valeur de n_2.

b) Modifier l'algorithme pour qu'il détermine le rang n_3 à partir duquel les termes de la suite seront inférieurs à 10^{-6}.

c) Quelle semble être la limite de la suite (u_n) lorsque l'entier n tend vers +∞ ?

Exercice IV : (Variables aléatoires)

On s'intéresse à un jeu qui consiste à lancer une fois un dé équilibré à 10 faces numérotées de 1 à 10.
La fonction python ci-dessous permet de simuler le gain en euros obtenu en jouant à ce jeu. Celui-ci est associé à la variable aléatoire X :

```python
from random import randint

def de10():
    face_tirée = randint(1,10)
    if 1 <= face_tirée <= 3:
        X = -2
    else:
        X = 5
    return(resultat)
```

1) a) Quelles sont les valeurs prises par la variable aléatoire X ?

b) Détailler les événements {X = -2} et {X = 5}.

c) En déduire la loi de probabilité de X.

2) a) On considère la fonction Simul(n) suivante :

```python
def Simul(n):
    S=0
    for i in range(n):
        S = S + de10()
    return(S/n)
```

b) Exécuter Simul(100), Simul(1000). Pouvait-on prévoir ces résultats ? Justifier.

Exercice V : (Statistiques seconde)

Les 1ère D ont eu les résultats d'une évaluation donnée par le professeur de mathématiques. Les différentes notes obtenues par les élèves ont été consignées dans une liste L1 et les effectifs de ces notes dans une liste E1.

*1) Compléter le programme python de la fonction **moypond** pour qu'elle retourne la moyenne des notes rentrées dans le paramètre L pondérées par les effectifs rentrés dans le paramètre E.*

```python
L1 = [6,7,8,10,11,12,13,14,16,18]
E1 = [1,1,2,5,8,4,3,1,2,2]

def moypond(L,E):
    n = len(L)
    Somme = 0
    Efftot = 0
    for i in range(n):
        Somme = Somme + ...
        Efftot = Efftot + ...
    Moyennep = ...
    return(Moyennep)
```

2) Quel serait le résultat retourné si on exécutait la commande
`>>> moypond(L1,E1)` *?*

3) Exécuter ce programme et vérifier votre réponse.

<u>Exercice VI</u> : (statistiques seconde)

On considère la liste L1 écrite dans un programme python :

```
>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

1) a) Quel est l'indice de la valeur 6 dans la liste L1 ? Expliquer.

b) Quelle ligne de commande, en python, doit-on écrire pour afficher la $6^{\text{ème}}$ valeur de la liste ?

c) Quelle ligne de commande, en python, doit-on écrire pour afficher la demi-somme de la $5^{\text{ème}}$ et de la $6^{\text{ème}}$ valeur.

2) Soit n un entier naturel. On souhaite générer une liste de n notes sur 20 avec un programme python.

a) Le programme ci-dessous répond-il à la question ? Expliquer.

```
from random import randint

def Listenote(n):
    L = [randint(0,20) for i in range(n)]
    return(L)
```

b) Utiliser le programme ci-dessus pour générer une liste de 10 notes sur 20 qu'on appellera L2 et une liste L3 de 11 notes sur 20.

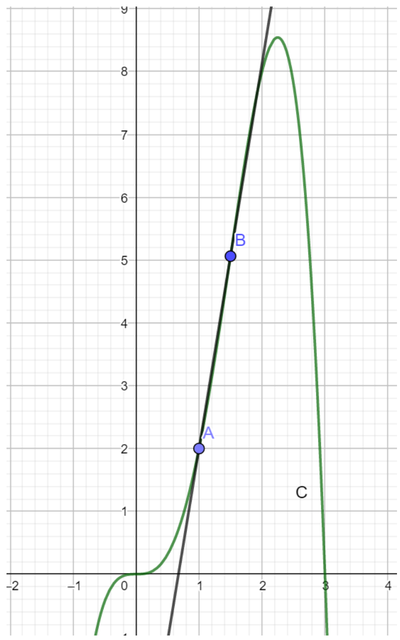
3) a) Compléter le programme ci-dessous pour que la fonction Med qui prend en paramètre une liste de valeurs, renvoie la médiane de la série de valeurs correspondante.

```
def Med(L):
    n = len(L)
    Lt = sorted(L)
    p = n//2
    if n%2 == 0:
        mediane = .........
    else:
        mediane = .........
    return(mediane)
```

La fonction sorted(L) permet de trier les valeurs de la liste L en ordre croissant.

b) A l'aide de la fonction Med, déterminer les médianes des séries L2 et L3 créées à la question 2.

Exercice VII : (Dérivation)



On considère la fonction f définie sur ℝ par : $f(x) = -x^4 + 3x^3$.
*On note **C** sa courbe représentative.*

1) a) Compléter le programme de la fonction f en langage python :

```
def f(x):
    return(.............)
```

b) Tester cette fonction en calculant f(1) et f(1,5).

*2) a) Calculer la pente de la sécante à **C** passant par les points A et B d'abscisses respectives 1 et 1,5.*

*b) Déterminer la pente de la sécante à **C** passant par les points A et M(1+h ; f(1+h)) avec h un réel non nul.*

*c) Recopier et compléter le programme python ci-dessous qui permet de créer la liste des pentes des sécantes à **C** passant par les points 1 et 1+h, avec h un réel qui varie de 0,1 à 0,01 avec un pas de 0,01.*

```
L = []
h = 0.1
while h > 0.01:
    m = ...........
    L = L + [m]
    h = .......
print(L)
```

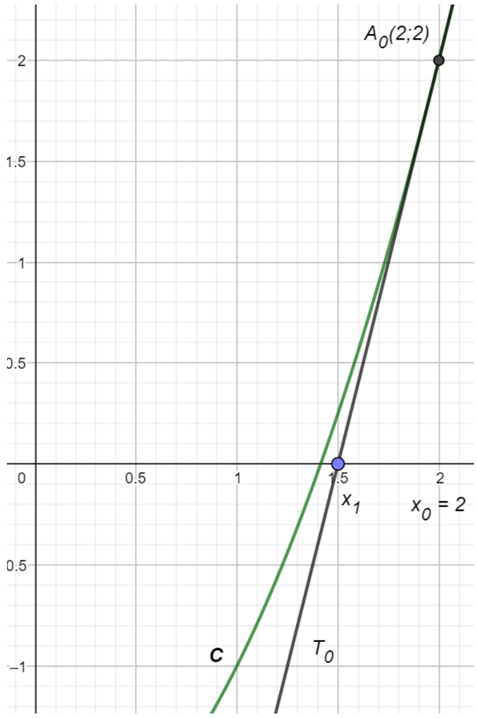
Que vaut la liste L à la fin de l'exécution du programme ?

d) Quel semble être la valeur du nombre dérivée de f en 1 ?

e) Calculer f'(1) et vérifier votre conjecture.

<u>Exercice VIII</u> : (Dérivation, suites)

*On considère la fonction f définie sur $\mathbb{R}$ par : $f(x) = x^2 - 2$ et **C** sa courbe représentative donnée ci-contre.*
On sait que l'équation $f(x) = 0$ admet $\sqrt{2}$ comme unique solution dans l'intervalle [1 ; 2].
Le but de cet exercice est de déterminer avec un programme en python une valeur approchée de $\sqrt{2}$ avec la méthode de Newton.
*Cette méthode consiste à tracer successivement des tangentes à la courbe **C** en procédant de la manière suivante :*



*1) a) Déterminer l'équation de la tangente T_0 à **C** au point A_0 d'abscisse $x_0 = 2$.*

b) Déterminer x_1 l'abscisse du point d'intersection de T_0 avec l'axe des abscisses.

*2) a) Déterminer l'équation de la tangente T_1 à **C** au point A_1 d'abscisse x_1.*

b) Montrer que x_2 l'abscisse du point d'intersection de T_1 avec l'axe des abscisses est égal à $\frac{17}{12}$.

3) On réitère le procédé n fois (avec n entier naturel).

*a) Déterminer en fonction de x_n l'équation de la tangente T_n à **C** au point A_n d'abscisse x_n.*

b) Si x_{n+1} est l'abscisse du point d'intersection de T_n avec l'axe des abscisses, montrer qu'on a la relation :

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right)$$

La relation de récurrence précédente permet de définir une suite (x_n) qui tend vers $\sqrt{2}$ quand n tend vers $+\infty$.

4) a) Compléter le programme python suivant pour qu'il calcule le $N^{ième}$ terme de la suite (x_n):

```
def SuiteX(N):
    x = 2
    n = 0
    while n < N:
        x = ........
        n = ........
    return(x)
```

b) Exécuter SuiteX(5). A quoi correspond la valeur obtenue ?