Python

Exercice 1

On considère la suite numérique (u_n) définie par son premier terme $u_0 = 2$ et pour tout entier naturel n, par :

$$u_{n+1} = \frac{2u_n + 1}{u_n + 2}$$

On admet que la suite (u_n) est bien définie, décroissante et a pour limite 1.
On considère le programme suivant écrit en langage Python :

```
1  def algo(p):
2    u=2
3    n=0
4    while u-1>p:
5      u=(2*u+1)/(u+2)
6      n=n+1
7    return (n,u)
```

1. Interpréter les valeurs `n` et `u` renvoyées par l'appel de la fonction `algo(p)` dans le contexte de l'exercice.
2. Donner, sans justifier, la valeur de n pour $p = 0,001$.

Exercice 2

Un patient doit prendre toutes les heures une dose de 2 ml d'un médicament.
On introduit la suite (u_n) telle que le terme u_n représente la quantité de médicament, exprimée en ml présente dans l'organisme immédiatement après n prises de médicament.
On a $u_1 = 2$ et

$$\text{pour tout entier naturel } n \text{ strictement positif : } u_{n+1} = 2 + 0,8u_n.$$

En utilisant la même modélisation, le médecin s'intéresse à la quantité moyenne de médicament présente dans l'organisme du malade au cours du temps.
On définit pour cela la suite (S_n) définie pour tout entier naturel n strictement positif par

$$S_n = \frac{u_1 + u_2 + \cdots + u_n}{n}.$$

On admet que la suite (S_n) est croissante et que pour tout $n \in \mathbb{N}^*$,

$$u_1 + u_2 + \cdots + u_n = 10n - 40 + 40 \times 0,8^n.$$

1. Calculer $\lim\limits_{n\to+\infty} S_n$.
2. On donne la fonction mystère suivante, écrite en langage Python :

```
1  def mystere(k) :
2      n = 1
3      s = 2
4      while s < k :
5          n = n + 1
6          s = 10 - 40/n + (40*0.8**n)/n
7      return n
```

   Dans le contexte de l'énoncé, que représente la valeur renvoyée par la saisie `mystere(9)` ?
3. Donner la valeur renvoyée par la saisie `mystere(9)`.

Exercice 3

Partie A

Soit (u_n) la suite définie par $u_0 = 30$ et, pour tout entier naturel n, $u_{n+1} = \frac{1}{2}u_n + 10$.

Soit (w_n) la suite définie pour tout entier naturel n par :

$$\begin{cases} w_0 = 45 \\ w_{n+1} = \frac{1}{2}w_n + \frac{1}{2}u_n + 7 \end{cases}$$

1. Montrer que $w_1 = 44,5$.

   On souhaite écrire une fonction suite, en langage Python, qui renvoie la valeur du terme w_n pour une valeur de n donnée. On donne ci-dessous une proposition pour cette fonction suite.

```
1  def suite(n) :
2      U=30
3      W=45
4      for i in range (1,n+1) :
5          U=U/2+10
6          W=W/2+U/2+7
7      return W
```

2. L'exécution de suite(1) ne renvoie pas le terme w_1. Comment modifier la fonction suite afin que l'exécution de suite(n) renvoie la valeur du terme w_n ?

Exercice 4

Partie A

On considère la fonction f définie sur l'intervalle $]-1\ ;\ +\infty[$ par

$$f(x) = 4\ln(x+1) - \frac{x^2}{25}$$

On considère h la fonction définie sur l'intervalle $[2\,;\,6{,}5]$ par $h(x) = f(x) - x$.
On donne ci-dessous le tableau de variations de la fonction h :

x	2		$m \approx 2{,}364$		6,5
$h(x)$	$h(2)$	↗	$M \approx 2{,}265$	↘	$h(6{,}5)$

1. Montrer que l'?équation $h(x) = 0$ admet une unique solution $\alpha \in [2\ ;\ 6{,}5]$.
2. On considère le script suivant, écrit en langage Python :

```
from math import *

def f(x) :
    return 4*log(1+x)-(x**2)/25

def bornes(n) :
    p = 1/10**n
    x = 6
    while f(x)-x > 0 :
        x = x + p
    return (x-p, x)
```

On rappelle qu'en langage Python :
- la commande log(x) renvoie la valeur $\ln x$;
- la commande c**d renvoie la valeur de c^d.

(a) Donner les valeurs renvoyées par la commande bornes(2).
On donnera les valeurs arrondies au centième.

(b) Interpréter ces valeurs dans le contexte de l'exercice.

Exercice 5

On se propose de comparer l'évolution d'une population animale dans deux milieux distincts A et B.
Au 1er janvier 2025, on introduit 6000 individus dans chacun des milieux A et B.

- On suppose que dans le milieu A, l'évolution de la population est modélisée par une suite géométrique (u_n) de premier terme $u_0 = 6$ et de raison $0,93$.
  Pour tout entier naturel n, u_n représente la population au 1er janvier de l'année $2025+n$, exprimée en millier d'individus.
  On admet que pour tout $n \in \mathbb{N}$, on a $u_n = 6 \times 0,93^n$
- On suppose que dans le milieu B, l'évolution de la population est modélisée par la suite (v_n) définie par
  $$v_0 = 6 \text{ et pour tout entier naturel } n,\ v_{n+1} = -0,05v_n^2 + 1,1v_n.$$
  Pour tout entier naturel n, v_n représente la population au 1er janvier de l'année $2025+n$, exprimée en millier d'individus.

1. Calculer u_1 et v_1.

2. On admet qu'à partir d'une certaine année, la population du milieu B dépassera la population du milieu A.On considère le programme Python ci-dessous.

```
n=0
u = 6
v = 6
while ... :
    u = ...
    v= ...
    n = n+1
print (2025 + n)
```

(a) Recopier et compléter ce programme afin qu'après exécution, il affiche l'année à partir de laquelle la population du milieu B est strictement supérieure à la population du milieu A.

(b) Déterminer l'année affichée après exécution du programme.

Exercice 6

On considère la suite (I_n) définie pour tout entier naturel n par :

$$I_n = \int_0^1 x^n \, \mathrm{e}^{-x} \, \mathrm{d}x.$$

On admet que l'on a :

$$I_0 = 1 - \frac{1}{\mathrm{e}} \qquad \text{et} \qquad \forall n \in \mathbb{N}, I_{n+1} = (n+1)I_n - \frac{1}{\mathrm{e}}.$$

On donne ci-dessous le script de la fonction `mystere`, écrite en langage Python.
On a importé la constante e.

```
def mystere(n):
    I = 1 - 1/e
    L = [I]
    for i in range(n):
        I = (i + 1)*I - 1/e
        L.append(I)
    return L
```

Que renvoie `mystere(100)` dans le contexte de l'exercice ?

Exercice 7

Une donnée binaire est une donnée qui ne peut prendre que deux valeurs : 0 ou 1.
Une donnée de ce type est transmise successivement d'une machine à une autre.
Chaque machine transmet la donnée qu'elle détient soit de manière fidèle, c'est-à-dire en transmettant l'information telle qu'elle l'a eue (1 devient 1 et 0 devient 0), soit de manière contraire (1 devient 0 et 0 et devient 1).
La transmission est fidèle dans 90 % des cas, et donc contraire dans 10 % des cas.

Dans tout l'exercice, la première machine récupère toujours la valeur 1.

Pour modéliser en langage Python la transmission de la donnée binaire décrite en début d'exercice, on considère la fonction `simulation` qui prend en paramètre un entier naturel n qui représente le nombre de transmissions réalisées d'une machine à une autre, et qui renvoie la liste des valeurs successives de la donnée binaire.
On donne ci-dessous le script incomplet de cette fonction.
On rappelle que l'instruction `rand()` renvoie un nombre aléatoire de l'intervalle $[0\,;\,1[$.

```
1  def simulation(n):
2      donnee = 1
3      liste = [donnee]
4      for k in range(n):
5          if rand() < 0.1 :
6              donnee = 1 - donnee
7          liste.append(donnee)
8      return liste
```

Par exemple, `simulation(3)` peut renvoyer [1, 0, 0, 1].Cette liste traduit :

- qu'une donnée binaire a été successivement transmise trois fois entre quatre machines ;
- la première machine qui détient la valeur 1 a transmis de manière contraire cette donnée à la deuxième machine ;
- la deuxième machine a transmis fidèlement la donnée qu'elle détient à la troisième ;
- la troisième machine a transmis de manière contraire la donnée qu'elle détient à la quatrième.

1. Déterminer le rôle des instructions des lignes 5 et 6 de l'algorithme ci-dessus.
2. Calculer la probabilité que simulation(4) renvoie la liste [1, 1, 1, 1, 1] et la probabilité que `simulation(6)` renvoie la liste [1, 0, 1, 0, 0, 1, 1].

Exercice 8

On considère la suite u définie sur $\mathbb{N}$ par :

$$u_0 = 6 \qquad \text{et} \qquad \forall n \in \mathbb{N}, u_{n+1} = \sqrt{3u_n - 2}$$

On admet que la suite (u_n) est bien définie et :

$$2 \leqslant u_{n+1} \leqslant u_n \leqslant 6$$

D'autre part, on admet que la suite u converge vers un réel $\ell = 2$

On considère la fonction `rang` écrite ci-dessous en langage Python.
On rappelle que sqrt(x) renvoie la racine carrée du nombre x.

```
1  from math import *
2
3  def rang(a) :
4    u = 6
5    n=0
6    while u > = a :
7      u = sqrt(3*u - 2)
8      n = n+1
9    return n
```

1. Pourquoi peut-on affirmer que `rang`(2.000001) renvoie une valeur ?
2. Pour quelles valeurs du paramètre a l'instruction `rang`(a) renvoie-t-elle un résultat ?

Exercice 9

Pour tout x appartenant à l'intervalle $]0\,;\,8]$, on note :

$$\mathcal{A}(x) = 10\ln\left(-x^2 + 7x + 9\right)$$

On admet que la fonction $\mathcal{A}$ est croissante sur $[0; 3,5]$ et décroissante sur $[3,5; 8]$.
On considère un réel strictement positif k.
On souhaite déterminer la plus petite valeur de x, approchée au dixième, appartenant à $[3{,}5\,;\,8]$ pour laquelle l'aire $\mathcal{A}(x)$ devient inférieure ou égale à k.
Pour ce faire, on considère l'algorithme ci-dessous.
Pour rappel, en langage Python, $\ln(x)$ s'écrit log (x).

```
1   from math import *
2
3   def A(x) :
4     return 10*log (- 1* x**2 + 7*x + 9)
5
6   def pluspetitevaleur(k) :
7     x = 3.5
8     while A(x)..... :
9       x = x + 0.1
10    return ...........
```

1. Recopier et compléter les lignes 8 et 10 de l'algorithme.
2. Quel nombre renvoie alors l'instruction `pluspetitevaleur`(30) ?
3. Que se passe-t-il lorsque $k = 35$? Justifier.

Exercice 10

Un étudiant mange tous les jours au restaurant universitaire. Ce restaurant propose des plats végétariens et des plats non végétariens.

Pour tout entier naturel n non nul, on note V_n l'évènement « l'étudiant a choisi un plat végétarien le n^{e} jour » et p_n la probabilité de V_n.
Le jour de la rentrée, l'étudiant a choisi le plat végétarien. On a donc $p_1 = 1$.

On admet que pour tout $n \geqslant 1$, $p_{n+1} = 0,2p_n + 0,7$.

On souhaite disposer de la liste des premiers termes de la suite (p_n) pour $n \geqslant 1$.
Pour cela, on utilise une fonction appelée `repas` programmée en langage Python dont on propose trois versions, indiquées ci-dessous.

Programme 1

```
1  def repas(n):
2    p=1
3    L=[p]
4    for k in range(1,n):
5      p = 0.2*p+0.7
6      L. append(p)
7     return(L)
```

Programme 2

```
1  def repas(n):
2    p=1
3    L=[p]
4    for k in range(1,n+1):
5      p = 0.2*p+0.7
6      L. append(p)
7    return(L)
```

Programme 3

```
1  def repas(n):
2    p=1
3    L=[p]
4    for k in range(1,n):
5      p = 0.2*p+0.7
6      L.append(p+1)
7    return(L)
```

1. Lequel de ces programmes permet d'afficher les n premiers termes de la suite (p_n) ? Aucune justification n'est attendue.
2. Avec le programme choisi à la question **a.** donner le résultat affiché pour $n = 5$.

Exercice 11