HEC 2009

Voie E

Problème (Enoncé)

Partie I. Analyse et algorithmique

Dans toute cette partie, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 = 0$, $u_1 = 1$ et l relation :

$$\forall n \in \mathbb{N},\ u_{n+2} = u_{n+1} + u_n$$

1. a) Compléter la fonction Python suivante pour qu'elle renvoie $u_0, u_1, \dots, u_n$:

```
import numpy as np
def u(n):
    v=np.zeros(n)
    v[1]=1
    ..........
    ..........
    return v
```

   b) On exécute l'instruction `S=np.cumsum(u(100))`. Que contient la variable `S` ?

2. a) Montrer que la suite $(u_n)_{n\in\mathbb{N}}$ est une suite croissante d'entiers naturels.

   b) La suite $(u_n)_{n\in\mathbb{N}}$ est-elle convergente ?

3. a) On note a et b les solutions de l'équation $x^2 - x - 1 = 0$, avec $a > b$. Justifier que :

$$b = 1 - a = -\frac{1}{a}$$

   Établir également l'encadrement : $1 < a < 2$.

   b) Montrer que :

$$\forall n \in \mathbb{N},\ u_n = \frac{1}{\sqrt{5}}\left(a^n - b^n\right)$$

   c) En déduire un équivalent de u_n lorsque n tend vers $+\infty$.

4. On pose :

$$\forall n \in \mathbb{N},\ \beta_n = u_{n+1} - a u_n$$

   Exprimer, pour tout $n \in \mathbb{N}$, β_n en fonction de n et de b.

5. On rappelle que la partie entière d'un réel x est l'entier noté $\lfloor x \rfloor$ vérifiant : $\lfloor x \rfloor \leqslant x < \lfloor x \rfloor + 1$.

   a) Établir, pour tout $n \in \mathbb{N}$, l'égalité suivante : $\lfloor a u_{2n} \rfloor = u_{2n+1} - 1$.

   b) Exprimer, pour tout $n \in \mathbb{N}^*$, $\lfloor a u_{2n-1} \rfloor$ en fonction de u_{2n}.

6. Soient y un réel fixé tel que $|y| < 1$ et k un entier fixé de $\mathbb{N}$.

   a) Montrer que la série $\sum n^k y^n$ est absolument convergente.

   b) En déduire la convergence de la série $\sum n^k \frac{u_n}{2^{n+1}}$.

c) Compléter le programme Python suivant pour qu'il calcule et affiche la valeur de $\sum_{n=0}^{N} n^2 \frac{u_n}{2^{n+1}}$ pour une valeur de N entrée par l'utilisateur :

```
import numpy as np
N=int(input('Entrez N : ')
   )
n=np.arange(0,N+1)
v=(n**2)*u(N)/2**(n+1)
print(..........)
```

d) En utilisant la définition de la suite $(u_n)_{n\in\mathbb{N}}$, calculer $\sum_{n=1}^{+\infty} \frac{u_n}{2^{n+1}}$.

Partie II. Probabilité et algorithmique

On effectue dans une urne contenant des boules numérotées 0 ou 1 une suite illimitée de tirages avec remise d'une boule. On suppose que l'urne contient une proportion p de boules numérotées 1 et une proportion q de boules numérotées 0, où p et q sont deux réels strictement positifs tels que $p + q = 1$. On admet qu'il existe un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ permettant de modéliser cette expérience et on suppose que les tirages sont mutuellement indépendants.
On s'intéresse au nombre de tirages nécessaires pour obtenir deux boules numérotées 1 de suite, c'est-à-dire lors de deux tirages consécutifs. On définit, pour tout $n \in \mathbb{N}^*$, l'événement S_i : « le i-ième tirage donne une boule numérotée 1 » et $B_i = S_i \cap S_{i+1}$.
Si au moins un des événements B_i se réalise au cours de l'expérience, on note Y la valeur de l'entier i correspondant au premier événement B_i réalisé. Sinon, c'est-à-dire si aucun des événements B_i se réalise, on attribue à Y la valeur 0. On admet que Y est une variable aléatoire sur $(\Omega, \mathcal{A}, \mathbb{P})$.
Par exemple, si le résultat de l'expérience est $0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, \ldots$, alors Y prend la valeur 6.

7. a) Compléter la fonction suivante pour qu'elle permette de simuler l'expérience et de renvoyer la valeur prise par Y.

```
import numpy.random as rd
def Simul(p):
    T=rd.random(2)
    i=1
    while T[0]>p or T[1]>p:
        T[0]=..........
        T[1]=..........
        i=............
    return i
```

b) On propose la fonction suivante :

```
def M(n,p):
    Y=np.zeros(n)
    for k in range(0,n):
        Y[k]=Simul(p)
    return np.mean(Y)
```

Que fait cette fonction ?

c) On propose le script suivant :

```
import numpy as np
n=int(input('n='))
p=float(eval(input('p=')))
print(np.mean(M(n,p)))
```

Une exécution de ce script avec $n = 1000$ et $p = \frac{1}{2}$ a renvoyé la valeur 4.947. Que peut-on conjecturer ?

8. a) Pour tout $i \in \mathbb{N}^*$, calculer $\mathbb{P}(B_i)$.

   b) Déterminer $Y(\Omega)$. Calculer $\mathbb{P}(Y = 1), \mathbb{P}(Y = 2)$ et $\mathbb{P}(Y = 3)$.

9. Pour tout $n \in \mathbb{N}^*$, on note C_n l'événement : « lors des n premiers tirages, il n'apparaît jamais deux fois de suite une boule numérotée 1 ».

   On note également : $C_0 = \Omega$.

   a) Déterminer $\mathbb{P}(C_0), \mathbb{P}(C_1)$ et $\mathbb{P}(C_2)$.

   b) Établir :

$$\forall n \in \mathbb{N},\ \mathbb{P}(Y = n + 2) = p^2 q\, \mathbb{P}(C_n)$$

10. a) En considérant les résultats possibles des deux premiers tirages, montrer, pour tout entier n supérieur ou égal à 2, l'égalité :

$$\mathbb{P}(C_n) = q\, \mathbb{P}(C_{n-1}) + pq\, \mathbb{P}(C_{n-2})$$

   b) Déterminer, pour tout $n \in \mathbb{N}^*$, une relation entre $\mathbb{P}(Y = n + 2), \mathbb{P}(Y = n + 1)$ et $\mathbb{P}(Y = n)$.

11. On suppose, dans cette question, que $p = q = \dfrac{1}{2}$.

   a) Montrer que :

$$\forall n \in \mathbb{N}^*,\ \mathbb{P}(Y = n) = \frac{u_n}{2^{n+1}}$$

   où la suite $(u_n)_{n\in\mathbb{N}^*}$ a été définie dans la première partie du problème.

   b) Que vaut $\mathbb{P}(Y = 0)$?

   c) Montrer que Y admet une espérance et la calculer. Comparer ce résultat avec la conjecture faite dans la question 7c.

   d) Montrer que Y admet une variance et la calculer.

12. On revient au cas général : $0 < p < 1$ et $q = 1 - p$ et on considère la fonction f définie par :

$$\forall x \in \mathbb{R},\ f(x) = x^2 - qx - pq$$

   a) Montrer que l'équation $f(x) = 0$ admet deux racines distinctes. On les note r et s, avec $r > s$.

   b) Établir les inégalités suivantes :

$$-1 < s < 0 < r < 1 \quad \text{et} \quad r > |s|$$

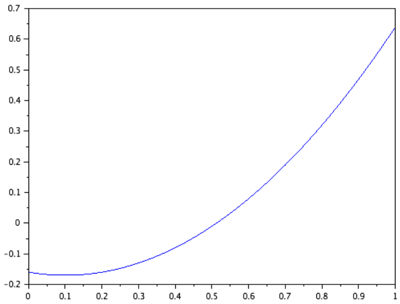
   c) Compléter la fonction Python suivante pour qu'elle renvoie la valeur de $f(x)$:

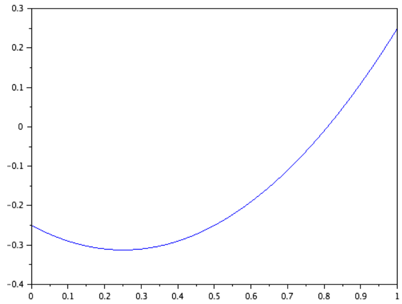
```
def f(p,x):
    return ..........
```

d) On exécute dans Python le script suivant avec $p = \frac{1}{2}$:

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0,1,81)
p=float(eval(input('Entrez p : ')))
y=f(p,x)
plt.plot(x,y)
```

- Le réel $\frac{1}{2}$ fait-il partie des nombres renvoyés par la commande `x=np.linspace(0,1,81)` ?
- Laquelle de ces deux courbes a été obtenue lorsqu'on a exécuté ce script ?





e) Compléter les commandes suivantes (qui s'ajouteront au programme précédent) pour qu'elles permettent de tracer la tangente à la courbe représentative de f au point d'abscisse r :

```
r=.....
z=.....
plt.plot(x,z)
```

13. a) On pose : $\Delta = q^2 + 4pq$. Montrer que :

$$\forall n \in \mathbb{N}^*, \ \mathbb{P}(Y = n) = \frac{p^2}{\sqrt{\Delta}} (r^n - s^n) \, .$$

b) Calculer $\mathbb{P}(Y = 0)$.

c) Montrer que Y admet des moments de tous ordres et calculer l'espérance de Y.