TP7 : Compléments - Concours

Exercice 1. (D'après EDHEC - voie S - 2016)

On considère la fonction f définie sur $\mathbb{R}_+^*$ par : $\forall x \in \mathbb{R}_+^*,\ f(x) = \frac{e^{-x}}{x}$.
On considère également la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 = 1$ et par la relation $u_{n+1} = f(u_n)$, valable pour tout entier naturel n. On admet que $(u_n)_{n\in\mathbb{N}}$ est bien définie. Les scripts suivants renvoient, pour celui de gauche, la valeur 5, et celui de droite, la valeur 6. Que sait-on de u_5 et u_6 ? Quelle conjecture peut-on émettre sur le comportement de la suite $(u_n)_{n\in\mathbb{N}}$?

```
u=1
n=0
while u>0.00001
      u = exp(-u)/u
      n=n+1
end
disp(n)
```

```
u=1
n=0
while u<100000
      u = exp(-u)/u
      n=n+1
end
disp(n)
```

Exercice 2. (D'après ECRICOME - voie S - 2016)

Soient a, b deux entiers strictement positifs. Une urne contient initialement a boules rouges et b boules blanches. On effectue une succession d'épreuves, chaque épreuve étant constituée des trois étapes suivantes :

- on pioche une boule au hasard dans l'urne,
- on replace la boule tirée dans l'urne,
- on rajoute dans l'urne une boule de la même couleur que celle qui vient d'être piochée

Après n épreuves, l'urne contient donc $a + b + n$ boules.
Pour tout $n \in \mathbb{N}^*$, on note X_n le nombre de boules rouges qui ont été **ajoutées** dans l'urne (par rapport à la composition initiale) à l'issue des n premières épreuves.
Pour tout $n \in \mathbb{N}^*$, on notera R_n l'événement « on pioche une boule rouge au n-ième tirage ».

1. Donner l'ensemble $X_n(\Omega)$ des valeurs prises par la variable aléatoire X_n en fonction de n.

2. On souhaite simuler l'expérience grâce à *Scilab*. Compléter la fonction suivante, qui simule le tirage d'une boule dans une urne contenant x boules rouges et y boules blanches et qui retourne la valeur 0 si la boule est rouge et 1 si elle est blanche.

```
function res = tirage(x,y)
        r = rand()
        if .......... then
                res = 0
        else
                res = 1
        end
endfunction
```

3. Compléter la fonction suivante, qui simule n tirages successifs dans une urne contenant initialement a boules rouges et b boules blanches (selon le protocole décrit ci-dessus) et qui retourne la valeur de X_n :

```
function Xn = experience(a,b,n)
        x = a
        y = b
        for k=1:n
                r = tirage(x,y)
                if r = = 0 then
                        x = ..........
                else
                        ...........
                end
        end
        Xn = .........
endfunction
```

Exercice 3. (D'après HEC Math I - voie S - 2016)

1. Écrire en *Scilab* une fonction « `function t=tr(A)` » qui calcule la trace d'une matrice carrée A.

2. La fonction « `issym` » suivante permet de tester si une matrice carrée A de taille n donnée est symétrique.

```
1 function b=issym(n,A)
2    b=%T; //affectation de la valeur booléenne True à la variable b.
3    for i=1:n-1
4       for j=i+1:n
5          b=b & A(i,j)==A(j,i)
6       end;
7    end;
8 endfunction
```

Préciser la signification de la ligne 5 du code et donner un exemple d'utilisation de la fonction « `issym` » en indiquant les valeurs d'entrée ainsi que la valeur de sortie obtenue.

3. La fonction « `orthoproj` » suivante, dont une ligne de code est incomplète, permet de tester si, pour une matrice carrée M de taille n donnée, M vérifie $tr(M)M^2 = tr(M^2)M$ et ${}^tM = M$.

```
1  function b=orthoproj(n,M)
2      A=tr(M)*M^2;
3      B=tr(M^2)*M;
4      b=issym(n,M);
5      if b then
6          for i=1:n
7              for j=i:n
8                  b=........
9              end;
10         end;
11     end;
12 endfunction
```

Compléter la ligne 8 du code et donner les valeurs de sortie obtenues par application de cette fonction aux deux matrices :

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Exercice 4. (D'après HEC Math II - voie S - 2016)

Soit V une variable aléatoire suivant la loi uniforme sur $[0, \pi/2[$. On pose $X = \tan^2(V)$. Compléter le code *Scilab* de la fonction `simulX` suivante de sorte que son application à l'entier $N (N \geqslant 2)$ fournisse une matrice colonne contenant N simulations indépendantes de la variable aléatoire X.

```
function x=simulX(N)
    u=rand(...,...);
    x=ones(u); //matrice de même format que u
    for i=1:N
        x(i,1)= ......
    end;
endfunction
```

Exercice 5. (D'après EDHEC - voie S - 2015)

Pour tout $n \in \mathbb{N}^*$, on pose $I_n = \displaystyle\int_1^{+\infty} \frac{\mathrm{d}x}{x^n(x+1)}$.

1. Montrer que I_n converge et calculer I_1.
2. Calculer, pour tout $n \in \mathbb{N}^*$, $I_n + I_{n+1}$.
3. Pour $n \in \mathbb{N}$, on pose $J_n = \displaystyle\int_1^{+\infty} \frac{\mathrm{d}x}{x^n(x+1)^2}$. Montrer que J_n converge et calculer J_0.
4. Pour tout $k \in \mathbb{N}^*$, exprimer $J_k + J_{k+1}$ en fonction de I_k.
5. À l'aide des questions précédentes compléter les commandes *Scilab* suivantes afin qu'elles permettent le calcul de I_n et J_n pour une valeur de n, supérieure ou égale à 2, entrée par l'utilisateur.

```
n=input("Entrez une valeur de n>=2 : ")
I=log(2); J=1/2; J=........
for k=2:n
    I=.........;  J=..........;
end
disp(I,"la valeur de I est :")
disp(J,"la valeur de J est :")
```

Exercice 6. (D'après EDHEC - voie S - 2015)

On considère une variable aléatoire X suivant la loi normale centrée et réduite. On pose $Y = |X|$ et on admet l'existence d'une variable aléatoire Z telle que la variable aléatoire $T = \sqrt{2Z}$ suivent la même loi que Y.
Écrire une commande *Scilab* permettant de simuler la variable aléatoire Z.

Exercice 7. (Inspiré d'ESSEC - voie S - 2003)

On rappelle (ou on admet) que tout entier naturel $n \in \mathbb{N}$ peut s'écrire en binaire (base 2). Autrement dit, pour tout $n \in \mathbb{N}$, il existe une unique liste $(a_p, a_{p-1}, \dots, a_0) \in \{0,1\}^{p+1}$ telle que :

$$n = a_p.2^p + a_{p-1}.2^{p-1} + \cdots + a_1.2^2 + a_1.2 + a_0$$

L'écriture binaire de n est alors $a_p a_{p-1} \dots a_1 a_0$ et on notera $\text{bin}(n) = (a_p, a_{p-1}, \dots, a_0)$.
Un exemple :
Pour tout entier $n \in [\![0, 31]\!]$, il existe une unique 5-liste $(a_4, a_3, a_2, a_1, a_0)$ telle que $n = a_4.2^4 + a_3.2^3 + a_2.2^2 + a_1.2 + a_0$.
Ainsi, avec $n = 22$, on peut trouver $22 = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0$.
L'écriture binaire de 22 est alors 10110 et $\text{bin}(22) = (1, 0, 1, 1, 0)$.

1. Déterminer l'écriture binaire de n, puis $\text{bin}(n)$ pour $n = 6$, $n = 15$ et $n = 21$.
2. On veut créer un programme `bin` qui prend en argument un entier naturel $n \in [\![0, 31]\!]$ et renvoie un vecteur `b` $= \text{bin}(n)$. Compléter-le.

```
function b=bin(n)
    b=[]
    for i=4:-1:0
        if 2^i<=n then
            b=...
            n=...
        else
            b=...
        end
    end
endfunction
```

3. Adapter le programme précédent pour qu'il renvoie `bin`(n) pour tout entier $n \in \mathbb{N}$.