Grenoble INP Ensimag	**PROBABILITES POUR L'INFORMATIQUE** **Apprentissage 2A** **2013-2014** **Durée : 2h** **Documents et matériel autorisés : polys d'exercices, notes de cours, poly de proba 1A, poly de stat 2A, poly de remise à niveau en maths, calculatrice.** Ce sujet comporte 2 pages d'énoncé

Ce devoir est composé de 3 exercices indépendants que vous pouvez traiter dans n'importe quel ordre.
Les résultats concernant les lois usuelles (densité, fonction de répartition de la loi normale, exponentielle, …) n'ont pas à être redémontrés.
Le barème est donné à titre indicatif, et est susceptible d'être modifié.
Il sera grandement tenu compte de la présentation et de la qualité de la rédaction (justification des réponses) dans la notation. Tout résultat manifestement erroné et non signalé entraînera la note de 0 à la question toute entière.

EXERCICE 1 Générateurs aléatoires (5 pts)

1. Justifier que le générateur congruentiel linéaire $x_{i+1} = 2x_i \mod 7$, avec $x_0 \neq 0$, n'est pas de pleine période.
2. Donner une valeur de $a > 0$ tel que le générateur congruentiel linéaire $x_{i+1} = ax_i \mod 7$, avec $x_0 \neq 0$, soit de pleine période.

EXERCICE 2 Étude de la loi d'un couple de variables aléatoires (7 pts)

On considère un couple de variables aléatoires réelles ayant pour densité $f_{X,Y}(x,y) = \begin{cases} Ce^{-2x}e^{-3y} & \text{si } 0 < y < x \\ 0 & \text{sinon.} \end{cases}$

où C est une quantité inconnue, à déterminer.

1. Déterminer la valeur de la constante C de sorte que $f_{X,Y}(x,y)$ soit une densité sur $\mathbb{R}^2$.
2. Déterminer la densité marginale de X.
3. Déterminer la loi de Y (nom de la loi et valeur du paramètre) et $E[Y]$ (sans calcul pour $E[Y]$).
4. Soit Z une variable aléatoire de loi exp(2). Montrer que $X|Y=y$ a même loi que $Z+K$ où K est une quantité déterministe que vous expliciterez. En déduire $E[X|Y=y]$.
5. En justifiant votre réponse mais sans faire aucun calcul d'intégrale ni de dérivée : étant donnés $y > 0$ et $x > 0$, donner la fonction de répartition de la loi de X sachant « $X > Y+x$ et $Y=y$ ».
6. X et Y sont-elles indépendantes ?
7. Donner un algorithme pour simuler le couple (X,Y).
8. Donner un algorithme pour simuler X.

Indication : si vous n'arrivez pas à démontrer un résultat ou à produire un algorithme, vous pouvez admettre que vous disposez de ce résultat ou cet algorithme dans les questions suivantes.

EXERCICE 3 Modélisation de la taille d'une pile (8 pts)

Dans un programme informatique, une procédure utilise une pile dont la taille X_n évolue de manière markovienne d'un pas n à l'autre du programme, cette taille étant toujours comprise entre 0 et 3. Au début du programme la taille de la pile est initialisée à 0. Les transitions de la chaîne de Markov sont décrites par le graphe Fig. 1.

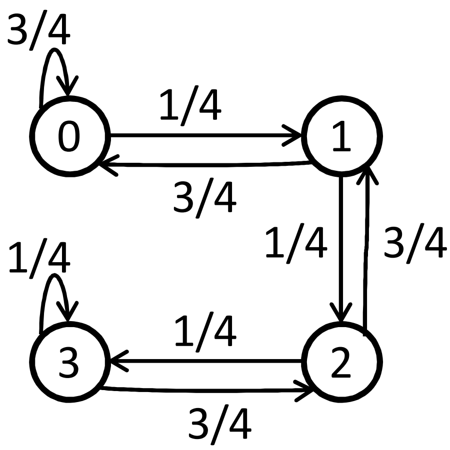

Fig. 1. Graphe de transition.

1. Donner la matrice de transition P de la chaîne de Markov.
2. Ecrire en C++ ou en pseudo-code une procédure de 7 lignes maximum (sans l'en-tête ni les déclarations de variables) qui fasse évoluer une pile telle que décrite ci-dessus pendant 100 pas d'exécution. On pourra supposer l'existence :
    a. D'une classe de piles *Stack* avec un constructeur Stack::Stack() qui produise une pile vide, et trois méthodes *Stack::push(Object)*, *Stack::pop()* et *Stack::size()* pour empiler, dépiler et récupérer la taille de la pile, où *Object* est un type ou une classe arbitraire (vous pouvez préciser le type retourné par les méthodes si vous le souhaitez).
    b. D'une fonction R() qui produise *true* avec probabilité ¼, *false* avec probabilité ¾.
3. Justifier qu'après un grand nombre de pas d'exécution du programme, l'espérance de la taille de la pile converge vers une quantité que vous calculerez.
4. On a considéré jusqu'ici que la taille initiale de la pile était nulle. On considère désormais qu'elle est de taille 0, 1, 2 ou 3 avec les probabilités respectives
$$\pi_0 = (\pi_0(0),...,\pi_0(3)).$$
Écrire une fonction ***simul*** en pseudo-code permettant de simuler la taille initiale de la pile de sorte qu'elle ait pour loi π_0. La fonction prendra en argument le vecteur π_0 et retournera une quantité aléatoire dont la loi sera π_0. On considère que l'on dispose d'une fonction ***runif*** qui simule des variables aléatoires indépendantes de loi uniforme sur [0,1].
5. Écrire une fonction en pseudo-code permettant de simuler la suite des tailles de la pile (dont la taille initiale aléatoire suit la loi π_0) en utilisant les probabilités de transition données dans l'énoncé. Cette fonction prendra en argument le nombre de pas d'exécutions n du programme, π_0 et les probabilités de transition (organisées sous forme de matrice P), et retournera une liste de tailles de piles simulées consécutivement. Toute simulation devra utiliser la fonction ***simul*** de la question 4 uniquement.
Indication : ce code devrait contenir 4 à 6 lignes.