T.D. 7 : Simulation (suite)

Exercice 1 (D'après EML 09)
Une urne contient des boules blanches et des boules noires. La proportion de boules blanches est 1/4 et la proportion de boules noires est 3/4.

On effectue des tirages successifs avec remise et on s'arrête dès que l'on a obtenu au moins une boule blanche et au moins une boule noire.
On note X la variable aléatoire égale au nombre de tirages effectués.
On note Y la variable aléatoire égale au nombre de boules blanches obtenues.
On note Z la variable aléatoire égale au nombre de boules noires obtenues.
Ainsi, on peut remarquer que la probabilité de l'événement $(Y = 1) \cup (Z = 1)$ est égale à 1.

On décide de coder l'événement <obtenir une boule blanche > par 1 et l'événement
< obtenir une boule noire > par 0.

Compléter le programme suivant, pour qu'il affiche les valeurs de X, Y et Z après cette expérience aléatoire.

```
program eml;
var boule, x, y, z : integer;
begin
randomize;
x:=0; y:=0; z:=0;
repeat
      x:= _______ ;
      if _________ then boule:=1
                        else boule:=0;
      if boule=1 then ______
                  else _______;
until ____ AND ____ ;
writeln('x vaut ',x); writeln('y vaut ',y); writeln('z vaut ',z); readln;
end.
```

Exercice 2 (ESSEC III 2006)
On se propose d'étudier des algorithmes de simulation de lois de probabilité, écrits en langage Pascal. La procédure randomize initialise le générateur de nombres aléatoires. Une fois celle-ci appelée, les appels successifs à la fonction random simulent la réalisation de variables aléatoires indépendantes de même loi de probabilité uniforme sur]0; 1[.
Des programmes demandés, on n'écrira que la partie « décisive » de l'algorithme.

Dans tout l'exercice, on désigne par :
_ $(\Omega ; T; P)$ un espace probabilisé,
_ U une variable aléatoire de Ω dans $\mathbb{R}$ de loi de probabilité uniforme sur]0; 1[,
_ X la variable aléatoire de Ω dans $\mathbb{R}$ dont on souhaite simuler des réalisations,
_ p un réel strictement compris entre 0 et 1,
_ n un entier naturel non nul.

Lois binomiales
1. (a) Ecrire un algorithme qui simule la réalisation d'une variable aléatoire X qui suit la loi de Bernoulli de paramètre p.
(b) Justifier que l'algorithme ci-dessous simule la réalisation d'une variable aléatoire X qui suit la loi binomiale de paramètres n et p.
```
x:=0;
for k:=1 to n do
begin
u:=random;
if u < p then x:=x+1;
end;
write(x);
```
Quel est le nombre d'appels à la fonction random ?

2. Soient p_1 et p_2 des réels strictement compris entre 0 et 1.
On effectue n épreuves de Bernoulli indépendantes, de paramètres p_1. Y désignant le nombre de succès lors de cette première série d'épreuves, on effectue Y épreuves de Bernoulli indépendantes, de paramètres p_2. On s'intéresse alors à la variable aléatoire X donnant le nombre de succès dans cette seconde série d'épreuves.
(a) Terminer le programme de simulation de X ci-dessous.
```
y:=0;
for k:=1 to n do
begin
u:=random;
if u < p1 then y:=y+1;
end;
x:=0;
for k:=1 to _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
...
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
write(x);
```