Python td 6 bis - Exercices : simulation de variables aléatoires

Exercice 1
Pour tout entier n strictement positif, on considère l'expérience suivante : on dispose de n urnes initialement vides, numérotées de 1 à n et d'un grand stock de boules que l'on dispose une à une dans ces urnes. Pour chaque boule, on choisit au hasard et de façon équiprobable l'urne dans laquelle la boule est déposée. On note X_n le rang du premier placement pour lequel une des urnes contiendra deux boules.
Ecrire une fonction Python intitulée `def X(n):` qui simule cette expérience aléatoire et affiche la valeur de $X(n)$.

Exercice 2
On considère une suite de variables aléatoires $(X_k)_{k\in\mathbb{N}}$ suivant toutes la loi $\mathcal{G}(p)$ et une variable aléatoire N suivant la loi de Poisson de paramètre λ. Soit S la variable aléatoire définie par $S = \sum_{k=0}^{N} X_k$, c'est-à-dire que

$$\forall \omega \ \in \Omega, \quad S(\omega) = \sum_{k=0}^{N(\omega)} X_k$$

Ecrire une fonction Python intitulée `def S(p,lambda):` qui simule la variable S.

Exercice 3
Un joueur réalise une suite de lancers indépendants d'une pièce. Cette pièce donne Pile avec la probabilité $p \in]0,1[$. On note N la variable aléatoire égale au rang d'apparition du premier Pile.
Si N prend la valeur n, le joueur place n boules numérotées de 1 à n dans une urne, puis il extrait une boule au hasard dans cette urne. On dit alors que le joueur a gagné si la boule obtenue porte un numéro impair, et qu'il a perdu si la boule porte un numéro pair.
Ecrire un programme Python qui simule cette expérience aléatoire, qui affiche les valeurs de N et de X, puis qui indique si le joueur a gagné ou perdu.

Exercice 4
Soit U et V deux variables aléatoires indépendantes, telles que U suit la loi uniforme (à densité) sur $[-3,1]$ et V suit la loi uniforme (à densité) sur $[-1,3]$.
Soit $p \in]0,1[$ et Z une variable aléatoire indépendante de U et V, telle que $Z(\Omega) = \{-1,1\}$, $P(Z=1) = p$ et $P(Z=-1) = 1-p$.
On note enfin la variable aléatoire X définie par :

$$\forall \omega \in \Omega, \quad X(\omega) = \begin{cases} U(\omega) \text{ si } Z(\omega) = 1 \\ V(\omega) \text{ si } Z(\omega) = -1 \end{cases}$$

Ecrire une fonction Python intitulée `def X(p):` qui simule la variable X.

Exercice 5
(QSP HEC 2022)
On considère la fonction Python suivante :

```
def simul(p):
   stop=0
   n=0
   while stop==0:
      A=rd.binomial(1,p)
      B=rd.binomial(1,p)
      n=n+2
      if A!=B :
          stop=1
   return (n,B)
```

On appelle cette fonction avec $p \in]0,1[$.
On considère une suite de variables $(X_k)_{k\in\mathbb{N}^*}$, mutuellement indépendantes, telles que pour tout $k \in \mathbb{N}^*$, X_k donne le résultat du k-ième appel de `rd.binomial(1,p)` s'il a lieu.

1. On note alors (N,Y) le couple de variables aléatoires renvoyé par l'appel de la fonction.
   Déterminer la loi de N et vérifier que $\sum_{k\in N(\Omega)} P(N=k) = 1$.
2. Déterminer la loi de Y.
3. Question subsidiaire : intérêt de cet algorithme ?