CY TECH	**ING1 - GMA** **2025-2026** **TD-TP1** **Simulation de variables aléatoires discrètes**

Exercice 1

Soit X une v.a.r. qui suit la loi de Bernoulli de paramètre p, $X \sim \mathcal{B}(p)$, et soit U une variable aléatoire qui suit la loi uniforme sur $[0,1]$, $U \sim \mathcal{U}(0,1)$.

1. Montrer que $P(U \in [0, 1-p]) = P(X=0)$ et que $P(U \in [1-p, 1]) = P(X=1)$.

2. En déduire un algorithme simple pour simuler X à partir de simulations de U.

Exercice 2

Soit X une v.a.r. discrète, qui prend les valeurs $a_1 < a_2 < ... < a_k < ..$ avec les probabilités $p_i = P(X = a_i)$. On considère également la v.a.r. $[0,1]$, $U \sim \mathcal{U}(0,1)$.

1. Que peut-on dire de l'ensemble $X(\Omega) = \{a_1, a_2, ... a_k, ... \}$?

2. Déterminer l'expression de la fonction de répartition F_X de X en fonction des p_i et des a_i.
On notera $f_i = F_X(a_i)$ dont on donnera l'expression.

3. Montrer que pour tout $k \geqslant 2$, on a :
$$P(f_{k-1} < U \leqslant f_k) = p_k$$
Que devient le cas $k = 1$?

4. Déduire de ce qui précéde un algorithme pour simuler X à partir de simulations de U.

Exercice 3 (TP)

1. Ecrire, en Python, une fonction permettant de générer un échantillon de taille N d'une loi de Bernoulli de paramètre p que l'on peut choisir.
La fonction doit avoir comme paramètre N et p. Elle doit afficher les moyennes et variances théoriques et empiriques. Elle doit également afficher un diagramme à bâtons donnant les probabilités théoriques et les fréquences empiriques.

2. Ecrire, en Python, une fonction permettant de générer un échantillon de taille N des tirages d'un dé non équilibré dont vous donnerez les probabilités des faces sous la forme d'un vecteur $Proba$.

La fonction doit avoir comme paramètre N et $Proba$. Elle doit afficher les moyennes et variances théoriques et empiriques. Elle doit également afficher un diagramme à bâtons donnant les probabilités théoriques et les fréquences empiriques.

Exercice 4 (TP)

1. Rappeler quelle sont les valeurs prises, et la loi de probabilité d'une v.a.r. X qui suit la loi de Poisson de paramètre λ, $X \sim \mathcal{P}(\lambda)$.
2. Afin de limiter les calculs coûteux, en particulier pour l'exponentielle et le factoriel, donner une relation de récurrence permettant de déterminer $P(X = k+1)$ en fonction de $P(X = k)$.
3. Mêmes questions pour $X \sim \mathcal{B}(n, p)$, loi binomiale de paramètres n et p.
4. Ecrire, en Python, des fonctions permettant de générer un échantillon de taille N d'une loi de Poisson et d'une binomiale.