Exercice 1

Soit n un entier naturel non nul, a un réel strictement supérieur à 1 et p un réel strictement compris entre 0 et 1.

On considère une action boursière sur n périodes. On suppose qu'au temps $t = 0$ la valeur de l'action v_t est $v_0 = 1$. À chaque instant $t \in \{1 \dots n\}$ la valeur v_t de l'action est multipliée par a avec probabilité p et divisée par a avec probabilité $q = 1 - p$.

On note X_n la valeur observée de l'action à la date $t = n$, et Z_n la variable aléatoire égale au nombre de hausses qu'a connue l'action sur les n périodes. On suppose que les événements successifs de hausse et de baisse sont mutuellement indépendants.

1. On suppose ici que $n = 2$.

a) Donner la loi de X_2, son espérance et sa variance.

b) On suppose que la valeur de X_2 est supérieure ou égale à 1. Quelle est la probabilité qu'il y ait eu une hausse lors de la première période ?

2. On revient au cas général.

a) Donner la loi de Z_n, son espérance et sa variance.

b) Exprimer en fonction de Z_n le nombre de baisses observées de l'action sur les n périodes.

c) En déduire que $Z_n = \dfrac{1}{2}(f(X_n) + b_n)$ où f est une fonction à préciser et b_n une constante dépendant de n à préciser également.

d) Dans cette question, $p = 1/2$ et supposons n impair. Montrer que $P(X_n \geq a) = \dfrac{1}{2}$.

Exercice 2

(*méthode de capture-recapture*)

Dans un lac donné, on souhaite estimer le nombre N de poissons qu'il contient. Pour ce faire, on capture un nombre M ($\leq N$!) de poissons dans le lac que l'on marque afin de les identifier. Ensuite, on les relâche dans le lac. Après avoir attendu un temps suffisamment long pour que les poissons se mélangent, on capture un nouvel échantillon n de poissons. On note X_N la variable aléatoire égale au nombre de poissons marqués dans l'échantillon de recapture.

On suppose que $n \leq M \leq N/2$

1. Identifier la loi de X_N.

2. On réalise une nouvelle pêche de n poissons, et on observe que X_N prend la valeur $k \in \{1 \dots n\}$. Il est raisonnable de penser que la valeur observée de X_n est la plus probable (sauf malchance !). On dit que k réalise le maximum de vraisemblance. Sous cette hypothèse, on cherche alors la valeur de N qui rend le nombre $u_N = P(X_N = k)$ le plus grand possible. Pour cela :

a) Simplifier le quotient $q_N = \dfrac{u_{N+1}}{u_N}$.

b) Donner une condition sur N en fonction de M, k pour que $q_N > 1$.

c) En déduire les variations de la suite $(u_N)_{N \geq 2M}$.

d) Déterminer la valeur la plus probable de N.

Exercice 3

(*Infomratique*).

Soit $n \geq 1$ et X une variable aléatoire d'espace image

$$X(\Omega) = \{k_1 < \cdots < k_n\}$$

dont la loi est donnée par :

$k =$	k_1	k_2	$\ldots$	k_n
$P(X=k)$	p_1	p_2	$\ldots$	p_n

où $p_1, \ldots, p_n$ sont des réels positifs de somme égale à 1.

On représente sous python la variable X par deux listes, la première étant : $\texttt{K} = [\texttt{P}[0], \ldots, \texttt{P}[\texttt{n}-1]]$ représentant $X(\Omega)$ par $\texttt{k}[\texttt{j}] = x_{j+1}$ (attention au décalage d'indice en python), et la seconde $\texttt{P} = [\texttt{P}[0], \ldots, \texttt{P}[\texttt{n}-1]]$ telle que $\texttt{P}[\texttt{j}] = p_{j+1}$ (même remarque).

On rappelle que la commande `rand()` tire un flottant au hasard dans $[0, 1[$.

Ainsi , si on réalise un grand nombre d'appels de la fonction `rand()`, on a un nuage de flottants uniformément répartis dans le segment $[0, 1[$.

Or, puisque la somme de p_i fait 1, une fraction environ égale à p_j de ces flottants se trouve dans l'intervalle $[a_{j-1}, a_j[$ si on note $a_0 = 0, a_1 = p_1, a_2 = p_1 + p_2$, et en général :

$$a_k = \sum_{j=1}^{k} p_j \quad \forall k \leq n.$$

1. Faire un schéma du segment $[0, 1[$ sur lequel apparaissent les segments $[a_k, a_{k+1}[$ et indiquer leur longueur sur ce schéma.
2. Compléter sur votre copie le script de la fonction `simuvar(K,P)` ci-dessous qui prend en entrée la loi d 'une variable aléatoire réelle X et retourne en sortie une des valeurs de X suivant la loi de X.

```python
def simuvar(K,P):
    """  K,P : deux listes de flottants de même longueur.
         Simule le comportement d'une VAR d'espace image K
         et de probas associées P. La fonction retourne en
         sortie le flottant K[j] avec proba P[j]
    """
    t = rand()
    a = 0
    for j in range(......):
        a += P[j]
        if t < a:






# -- fin
```