Exercice 1
Un archer tire sur des cibles situées à 20 m et à 50 m. Il effectue trois tirs en changeant de cible à chaque fois. La probabilité d'atteindre la cible à 20 m (resp. 50 m) est p (resp. q) avec $q < p$. On suppose les tirs indépendants. Il gagne le jeu s'il atteint deux cibles consécutivement. Par quelle cible a-t-il intérêt à commencer ?

Exercice 2
Une école recrute ses élèves dans trois zones.
Il y a deux fois plus de candidats dans la zone 1 que dans la zone 2, et deux fois plus de candidats dans la zone 2 que dans la zone 3.
Le taux de réussite au concours pour les candidats de la zone 1 est 25 %, pour ceux de la zone 2 il est de 35 % et pour ceux de la zone 3 il est de 30 %.

1. Déterminer les proportions des candidats issus des 3 zones.
2. Quel est le taux de réussite au concours d'entrée de cette école ?
3. Dans cette école, à quelles proportions d'élèves issus de chacune des 3 zones devons nous nous attendre ?
4. Un candidat est recalé. Calculer la probabilité qu'il vienne de la zone 1.

Exercice 3
Dans une population comptant deux fois plus de femmes que d'hommes, 5% des hommes et 25 femmes sur 10000 sont daltoniens. On choisit au hasard une personne daltonienne de cette population. Quelle est la probabilité que ce soit un homme ?
Simuler cette expérience par une fonction python *simul* sans argument.
Déterminer une approximation de cette probabilité par simulations informatiques.

Exercice 4
Un tireur atteint sa cible 3 fois sur 10. Combien de tirs doit-il effectuer pour atteindre au moins une fois sa cible avec une probabilité supérieure ou égale à $0,8$?
Coder une fonction `simul6()` qui renvoie le nombre d'essais avant d'atteindre la cible.
Coder en python une fonction `Freq(n)` qui renvoie la 12-liste des fréquences des événements : "la cible est atteinte en 0 coups",..., "la cible est atteinte en 10 coups", "la cible n'est pas atteinte en moins de 11 coups", avec n simulations.
Écrire une fonction `tirs(x,n)` qui retourne le nombre de tirs nécessaires pour atteindre la cible avec une proba supérieure ou égale à x en utilisant la liste renvoyée par `Freq(n)` ; cette fonction renverra 11 si le nombre de tirs nécessaires est supérieur ou égal à 11.

Exercice 5
Soit n un entier naturel non nul. On effectue n lancers indépendants d'une pièce pour laquelle la probabilité d'obtenir "face" est p avec $p \in]0,\ 1[$. On pose $q = 1 - p$.
Quelle est la probabilité qu'au cours de ces n lancers, face ne soit jamais suivi de pile ?

Exercice 6
Une urne contient 7 boules rouges et 3 boules noires. On tire successivement et sans remise 3 boules. Calculer $\mathbb{P}(A)$ et $\mathbb{P}(B)$ où $A =$ "les trois boules sont rouges" et $B =$ "deux boules qui se suivent n'ont pas la même couleur". Simuler cette expérience puis retrouver les valeurs de ces probabilités par un grand nombre de simulations.

Exercice 7
Un gardien d'immeuble est ivre 3 jours sur 5. Il dispose d'un trousseau de 10 clefs indiscernables pour rentrer chez lui. Lorsqu'il est sobre, il essaie les clés les unes après les autres, en mettant de côté les essais infructueux. Lorsqu'il a bu, il fait tomber le trousseau après chaque tentative, mélangeant ainsi les clés.

1. Un soir, il lui a fallu 8 tentatives pour ouvrir la porte. Quelle est la probabilité qu'il soit rentré ivre ?
2. Écrire une fonction python `tentatives_ivre` sans paramètre qui renvoie le nombre de tentatives si le gardien est ivre.
3. Écrire une fonction python `tentatives_sobre` sans paramètre qui renvoie le nombre de tentatives si le gardien est sobre.
4. Écrire une fonction python `simul` sans paramètre qui renvoie le nombre d'essais du gardien un jour donné ainsi que l'état du gardien (*True* pour ivre, *False* sinon).
5. Écrire une fonction python `approx1(k,n)` qui renvoie une approximation de la probabilité qu'il y a eu k tentatives, avec n simulations.
6. Écrire une fonction python `approx2(n)` qui renvoie une approximation de la probabilité que le gardien soit ivre sachant qu'il y a eu 8 tentatives, avec n simulations.

Exercice 8 (Chaîne de Markov traitée de façon non matricielle)
Soit A une pièce équilibrée et B une pièce dont la probabilité d'obtenir face est $\frac{2}{3}$. On choisit une des deux pièces au hasard et on la lance. Si on obtient face, on conserve la pièce qu'on vient de lancer, sinon on change de pièce pour le lancer suivant. On effectue une suite de lancers en appliquant pour chacun d'eux le protocole du premier.

1. Quelle est la probabilité d'obtenir face au $n^{\text{ème}}$ lancer ?
   Donner une signification à la limite de cette probabilité lorsque n tend vers $+\infty$.
2. Calculer la probabilité d'avoir changé de pièce au deuxième lancer.
3. Calculer la probabilité d'avoir des résultats différents aux deux premiers lancers.
4. On suppose que l'on a joué avec la pièce A au troisième lancer.
   (a) Quelle est la probabilité d'avoir joué avec la pièce B au deuxième lancer ?
   (b) Quelle est la probabilité d'avoir joué avec la pièce B au premier lancer ?
5. À l'aide d'une simulation informatique, retrouver le résultat de la question 1.

Exercice 9

On dispose de 7 urnes $U_0, \dots, U_6$. L'urne U_k contient k boules blanches et $6-k$ boules noires. On choisit une urne au hasard puis on effectue p tirages avec remise dans l'urne choisie ($p \geqslant 3$). On note B_i : "la $i^{\text{ème}}$ boule tirée est blanche."

1. Déterminer la probabilité p_1 que le premier tirage ait amené une boule blanche.
2. Calculer la probabilité p_2 que les deux premiers tirages aient amené une blanche.
3. Les événements B_1 et B_2 sont-ils indépendants ? Ce résultat était-il prévisible ?
4. Déterminer la probabilité que la troisième boule soit blanche sachant que les deux premières le sont.
5. Calculer la probabilité que l'urne choisie soit U_6 si les 3 $1^{\text{ères}}$ boules sont blanches ?
6. Écrire une fonction Python *simul7* de paramètre p qui renvoie le résultat de cette expérience sous la forme d'une p-liste de 0 (noire) et de 1 (blanche).
7. Écrire une fonction *approx1* de paramètres n, p, q qui renvoie une approximation de la probabilité d'obtenir au moins q boules noires à l'aide de n simulations.
8. Écrire une fonction *approx2* de paramètres n, p qui renvoie une approximation de la probabilité que la $p^{\text{ème}}$ boule soit blanche sachant que les précédentes le sont.
9. Écrire une fonction *approx3* de paramètres n, p qui renvoie une approximation de la probabilité que l'urne choisie soit U_6 sachant que les boules tirées sont blanches.
10. Écrire une fonction *moyenne* de paramètres n, p qui renvoie la moyenne du nombre de couples de blanches consécutives sur n simulations.

Exercice 10

Montrer que si deux événements sont incompatibles et indépendants alors l'un des deux est de probabilité nulle. Deux événements incompatibles de probabilités non nulles peuvent-ils être indépendants ? Conclure.

Exercice 11

Soit $p \in [0,1]$. On considère une pièce de monnaie qui, lorsqu'on la lance, amène un pile avec la probabilité p et un face avec la probabilité $1-p$. Une personne lance cette pièce plusieurs fois. Elle gagne dès qu'elle a obtenu deux piles de plus que de faces, elle perd dès qu'elle a obtenu deux faces de plus que de piles. Les tirages cessent dès que la personne gagne ou perd.

1. Les lancers sont-ils indépendants ?
2. Quelle est la probabilité de l'événement D_n : "la partie dure plus de $2n$ coups" ? On pourra relier $\mathbb{P}(D_n)$ et $\mathbb{P}(D_{n-1})$.
3. Quelle est la probabilité pour que la personne gagne en exactement $2k+2$ coups ?
4. Quelle est la probabilité pour que la personne gagne en au plus $2n+2$ coups ?
5. Quelle est la probabilité pour que la personne perde en au plus $2n+2$ coups ?
6. Calculer $\lim\limits_{n\to+\infty} \mathbb{P}(D_n)$ et interpréter cette valeur.
7. Écrire une fonction Python *Simulation11(p)* qui renvoie le résultat de ce jeu sous la forme d'une liste de 0 (pile) et de 1 (face).
8. Écrire une fonction Python *Approx1(n,p)* qui renvoie une approximation de la probabilité de gagner à ce jeu à l'aide de n simulations.
9. Écrire une fonction Python *Approx2(n,p)* qui renvoie une approximation de la probabilité de perdre sachant qu'il y a eu au moins 10 lancers avec n simulations.
10. Écrire une fonction Python *Estimation(n,p)* qui renvoie une estimation du nombre moyen de lancers à l'aide de n simulations.
11. Pour $i \in \mathbb{N}$, on définit E_i : "la partie dure exactement $2i$ lancers".
    Écrire une fonction Python de paramètres (n, p, i) qui renvoie la $(i+2)$-liste d'approximations des probabilités $\mathbb{P}(E_0), \dots, \mathbb{P}(E_i), \mathbb{P}(D_i)$.

Exercice 12

Un jeu consiste à lancer plusieurs fois une pièce équilibrée.
Pour gagner il faut obtenir deux piles de suite. Le jeu s'arrête alors.
On définit les événements suivants :
P_n : "le $n^{\text{ème}}$ lancer a lieu et donne pile",
F_n : "le $n^{\text{ème}}$ lancer a lieu et donne face",
G_n : "la partie est gagnée au $n^{\text{ème}}$ lancer" et $x_n = \mathbb{P}(G_n)$.

1. A-t-on $\overline{P_n} = F_n$? Les événements P_1, P_2 sont-ils indépendants ? Même question pour P_1, P_2, P_3. Même question pour $P_1, P_2, \dots, P_n$ avec $n \geqslant 3$.
2. Calculer x_1, x_2, x_3.
3. Justifier que pour $n \geqslant 3$, on a $G_n \subset F_1 \cup (P_1 \cap F_2)$ et $\mathbb{P}(G_n|F_1) = \mathbb{P}(G_{n-1})$. En déduire $\mathbb{P}(G_n \cap F_1)$ en fonction de x_{n-1}.
4. Obtenir une relation analogue avec $\mathbb{P}\big(G_n \cap (P_1 \cap F_2)\big)$.
5. En déduire que pour $n \geqslant 3$ on a $x_n - \frac{1}{2}x_{n-1} - \frac{1}{4}x_{n-2} = 0$. Déterminer x_n en fonction de n.
6. Calculer $y_n = \sum\limits_{k=2}^{n} x_k$. Que représente ce nombre ? Quel est sa limite lorsque n tend vers $+\infty$? Quelle interprétation intuitive peut-on donner à cette limite ?
7. Écrire en python une fonction `simul()` qui renvoie le nombre de lancers lors d'une simulation de ce jeu. En déduire une fonction `approx(n,p)` qui renvoie une approximation de y_n par p simulations.