Exercice 1 (Probabilité qu'un nombre prescrit d'événements se réalisent)
Soient A, B, C trois événements tels que : $\mathbb{P}(A) = \mathbb{P}(C) = 0,3$; $\mathbb{P}(B) = 0,25$; $\mathbb{P}(A \cap B) = \mathbb{P}(B \cap C) = 0,15$; $\mathbb{P}(C \cap A) = 0,2$ et $\mathbb{P}(A \cap B \cap C) = 0,08$.
Calculer les probabilités des événements suivants :

1. Deux des trois événements se produisent mais pas les trois.
   Indication : on commencera par exprimer cet événement en fonction de A, B et C.
2. Un seul des trois événements se produit.
   Indication : on commencera par exprimer cet événement en fonction de A, B et C.
3. Aucun des trois événements ne se produit.

Exercice 2 (Tirages sans remise avec condition d'arrêt)
On considère une urne contenant b boules blanches, v boules vertes et n boules noires. Le joueur tire une boule, si elle est verte il gagne, si elle est blanche il perd, si elle est noire il la met de côté et tire une deuxième boule en suivant le protocole précédent, et ainsi de suite.
On note B (resp. V, N) l'événement : “la première boule tirée est blanche (resp. verte, noire).”
On note G_n l'événement : “le joueur gagne avec initialement n boules noires”, et $u_n = \mathbb{P}(G_n)$.

1. Justifier que $\mathbb{P}_N(G_n) = \mathbb{P}(G_{n-1})$.
2. Montrer que (B, V, N) est un SCE.
3. Trouver une relation entre u_n et u_{n-1}.
4. Calculer u_0 puis u_1 et conjecturer la valeur de u_n pour tout $n \in \mathbb{N}$.
5. Démontrer cette conjecture.

Exercice 3 (Jeu de cartes avec deux as de cœur)
Un jeu de 32 cartes est truqué : une des cartes a été remplacée par un $2^{\text{ème}}$ as de cœur. On tire trois cartes simultanément. Quelle est la probabilité que l'on se rende compte de la supercherie ? Même question si l'on tire p cartes. À partir de combien de cartes a-t-on au moins une chance sur deux de s'apercevoir que le jeu est truqué ?
Simuler ce jeu sous la forme d'une fonction Python qui prend en argument le nombre p de cartes tirées et qui renvoie `True` si les deux as de cœur sont tirés et `False` sinon.
Estimer la probabilité de découvrir la supercherie à l'aide d'une fonction Python de paramètres n (nombre d'expériences simulées) et p (nombre de cartes tirées).

Exercice 4 (Tirages avec remise dans une urne bicolore)
Une urne contient 7 boules blanches et 3 boules noires. On tire successivement et avec remise quatre boules de l'urne.

1. Quelle est la probabilité pour que l'on obtienne dans cet ordre deux boules noires puis deux boules blanches ?
2. Quelle est la probabilité d'obtenir deux boules blanches exactement ?
3. Quelle est la probabilité d'obtenir au moins une boule blanche ?
4. Écrire une fonction Python qui simule ce jeu en renvoyant une 4-liste de 0 et de 1 en convenant que 0 correspond à la couleur blanche et 1 à la couleur noire.
   Estimer, à l'aide d'un grand nombre de simulations, les probabilités que les boules blanches et noires alternent, et qu'il y ait exactement 2 boules blanches.

Exercice 5 (Paradoxe de Monty Hall)
Le Monty Hall est un jeu télévisé dont la règle est la suivante. Le candidat est devant trois portes : derrière se trouvent une voiture et deux chèvres. Le candidat commence par choisir une porte sur les trois. Le présentateur ouvre alors, parmi les deux portes restantes, une porte cachant une chèvre. Le candidat peut alors ouvrir la porte restante ou revenir sur son premier choix. Quelle est la meilleure stratégie ?
Indication : pour simplifier le problème on supposera que le joueur a choisi la porte 3. On notera O_i l'événement : “le présentateur ouvre la porte i” et V_i l'événement : “la voiture se trouve derrière la porte i”. On pourra faire un arbre ou bien utiliser une formule de probabilité.
Effectuer une simulation informatique du jeu Monty Hall en langage Python et retrouver la conclusion de l'étude mathématique. L'instruction `randint(1,n)` renvoie au hasard un entier entre 1 et n.