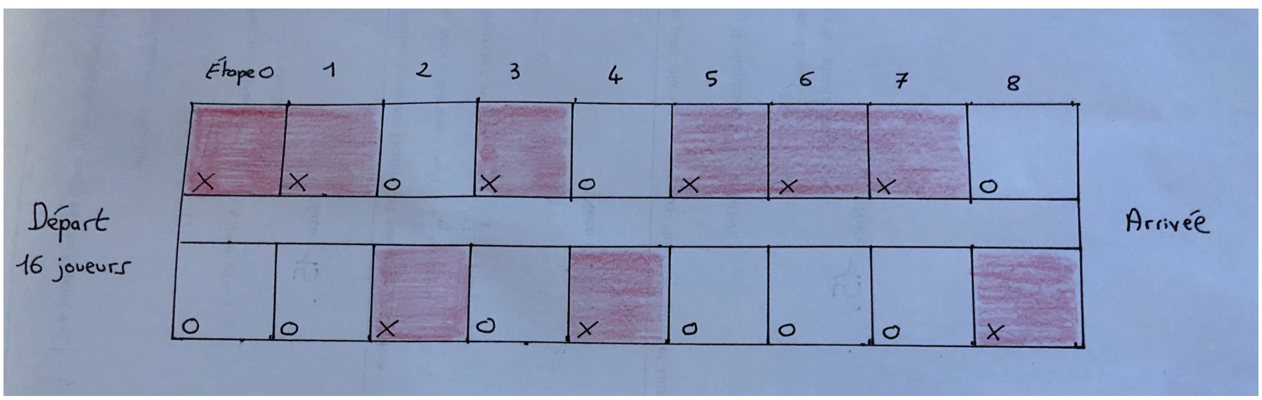
Ce devoir est à faire à deux. Les différentes fonctions à écrire sont à envoyer par mail le vendredi 12 novembre sous la forme d'un seul fichier Python. Pour les questions qui demandent une rédaction à l'écrit, vos copies sont à rendre le samedi 13 novembre.

Vos différentes fonctions devront être commentées et documentées. Des valeurs numériques pour tester vos fonctions sont suggérées dans le sujet, vous pouvez les prendre plus ou moins importantes, selon la puissance de l'ordinateur dont vous disposez. Vous pourrez commenter et tenter d'expliquer les résultats obtenus lors des tests de vos fonctions.

Principe du jeu

• Il y a 16 participants qui doivent franchir chacun leur tour un pont suspendu formé de dalles en verres. Il y a 18 étapes pour arriver de l'autre côté. À chaque étape, le participant doit choisir une dalle entre les deux proposées. L'une des deux est en verre trempé et ne se brisera pas sous le poids du joueur et l'autre se casse dès que l'on marche dessus. Dès que l'un des participants choisit une dalle fragile, il tombe dans le vide et son parcours s'arrête. Si par chance il a choisit une dalle solide, il passe à l'étape suivante et retente sa chance jusqu'à ce qu'il tombe ou qu'il arrive à la fin du parcours.

• Ci-dessous un schéma représentant la situation avec uniquement 9 dalles, numérotées de 0 à 8, au lieu des 18 qui correspondent à la situation que l'on va étudier (pour des raisons de place sur le dessin).



• Les dalles fragiles sont représentées avec le symbole × et les dalles solides avec le symbole ◯. Cependant les participants ne peuvent pas voir la différence entre les deux types de dalles à l'oeil nu et sont obligés de choisir une dalle au hasard.

• Il faut également comprendre que lorsqu'une dalle se brise et qu'un joueur tombe, le participant suivant a l'avantage de savoir sur quelle dalle il va falloir passer à cette étape-là. Inversement, on suppose aussi que si un participant choisit la bonne dalle à une certaine étape, le participant suivant (qui voit le parcours de tous les joueurs qui le précèdent) a une bonne mémoire et choisira aussi le bon parcours.

• Le but de l'exercice est programmer ce jeu en Python afin d'étudier les probabilités de survie des différents participants.

A-Survie du joueur 1

1. Donner la façon de créer une liste en compréhension, ***Ldalles***, contenant 18 nombres aléatoires qui valent soit 0 soit 1. Cette liste correspondra au choix aléatoire des dalles solides et des dalles fragiles lors des 18 étapes.
2. Le joueur 1 s'élance sur le parcours, il choisit au hasard un nombre, soit 0 soit 1. Si ce nombre est le même que le premier élément de la liste ***Ldalles***, il réussit l'étape 0 et passe à l'étape suivante pour tenter de deviner le bon numéro de l'étape 1. Ainsi de suite jusqu'à ce qu'il échoue ou qu'il arrive à deviner les 18 nombres de la liste ***Ldalles***. Écrire une fonction ***J1()*** qui renvoie le numéro de l'étape à laquelle le joueur 1 tombe, on rappelle que la numérotation des étapes commence à 0. La fonction renverra 18 si le joueur 1 a réussi à traverser le pont. Votre fonction ne renverra pas forcément le même résultat à chaque appel, puisqu'il y a un choix aléatoire.
3. Créer une fonction ***HistoJ1(N)*** qui prend en paramètre un entier naturel non nul N et trace l'histogramme des résultats obtenus en répétant N fois la fonction ***J1()***. Vous pourrez tester votre fonction avec $N = 10^6$.
4. (a) Créer une fonction ***DicoJ1(N)*** qui renvoie le dictionnaire dont les clés sont les numéros des étapes de 0 à 18 et la valeur correspondante est le nombre de fois où le joueur 1 s'est arrêté à l'étape en question lors de N répétitions du jeu.
   (b) Avec $N = 10^7$, combien de fois le joueur 1 s'est-il sauvé ?
   (c) Transformer le dictionnaire obtenu pour que la valeur associée à la clé $i \in [\![0, 18]\!]$ soit à présent le couple (a, b) avec a le nombre de fois où le joueur 1 s'est arrêté à l'étape i et b la probabilité qu'il s'arrête à l'étape i.
   (d) Est-ce que cela vous semble pertinent de représenter ces résultats sous la forme d'un dictionnaire ?
   (e) Justifier par un raisonnement mathématique les différentes probabilités observées.

B-Survie des 16 joueurs

1. Écrire une fonction ***Jeu()*** qui renvoie le numéro du premier participant qui réussit à franchir le pont suspendu. Les participants sont numérotés de 1 à 16. Cette fonction n'est pas simple à écrire et vous devez commenter votre script pour le rendre compréhensible. Bien entendu, votre fonction ne renvoie pas le même résultat à chaque appel puisqu'il y a l'aléa du choix des dalles. Il est possible, mais rare, que le joueur 16 ne soit pas sauvé, dans ce cas votre fonction renverra 17, 18 ou 19 en imaginant qu'il y a des joueurs fictifs supplémentaires (le 19-ème joueur étant toujours sauvé).
2. (a) Écrire une fonction ***Moy(N)*** qui répète le jeu précédent N fois et renvoie la moyenne des résultats obtenus.
   (b) Tester votre fonction avec $N = 10^6$, quelle est la valeur moyenne obtenue ?
   (c) À l'aide d'un raisonnement mathématique, expliquer cette valeur moyenne.
3. On remarque que si la fonction ***Jeu()*** renvoie le numéro j cela signifie que, lors de ce jeu, tous les joueurs ayant un numéro supérieur ou égal à j ont réussi à traverser le pont suspendu.
   (a) Écrire une fonction ***Resultats(N)*** qui répète le jeu N fois et renvoie une liste, L, de 16 éléments qui correspondent au nombre de fois où chaque joueur a réussi à traverser le pont. Par exemple, $L[2]$ est le nombre de fois où le joueur 3 a réussi l'épreuve.
   (b) Tester votre fonction avec $N = 10^6$, commenter les valeurs obtenues.
   (c) Tracer le graphe avec pour abscisses les 16 joueurs et en ordonnées le nombre de fois où le joueur a été sauvé, on prendra toujours $N = 10^6$.
4. (a) Modifier la fonction de la question 3.(a) pour renvoyer la liste des probabilités de survie de chaque joueur.
   (b) En quelles positions devez-vous franchir le pont pour avoir plus de 95% de réussite ?

C-Généralisation

On généralise le jeu avec NJ joueurs et ND dalles avec $(NJ, ND) \in (\mathbb{N}^*)^2$.

1. Généraliser vos fonctions précédentes pour renvoyer la liste des probabilités de succès de chaque joueur lorsque l'on répète N fois cette épreuve. On appellera ***Probas(N, NJ, ND)*** cette fonction.
2. (a) Tester votre programme avec NJ joueurs et le même nombre de dalles pour NJ allant de 1 à 5, toujours avec une répétition de 10^6 épreuves.

   (b) Écrire les probabilités obtenues sous la forme d'une fraction dont le dénominateur est 2^{NJ}. Bien entendu, les probabilités que vous obtenez avec votre programme sont des valeurs approchées mais elles sont suffisamment précises pour deviner la valeur exacte, comme demandé dans cette question.

   (c) Expliquer d'où viennent les numérateurs que vous observez. On pourra pour cela former le tableau représentant ces différentes probabilités comme ci-dessous. Le joueur 0 n'existe pas, c'est une convention pour faciliter la construction du tableau.

	Joueur 0	Joueur 1	Joueur 2	Joueur 3	Joueur 4	Joueur 5
1 dalle	0	1/2	1	1	1	1
2 dalles	0	1/4	3/4	1	1	1
3 dalles	0	1/8	4/8	7/8	1	1
4 dalles	0	1/16	5/16	11/16	15/16	1

   (d) Écrire sous la forme d'une somme de coefficients binomiaux le numérateur de la fraction qui correspond à la probabilité de survie du joueur 456 lors d'une épreuve avec 1000 dalles.
3. Tracer le graphe indiquant le temps mis pour exécuter la fonction ***Probas(10**5, NJ, NJ)*** en fonction du nombre de joueurs NJ. En déduire la complexité de votre algorithme.