

Exercice 1 : [Question barrière] Soit $n \geq 2$ un entier fixé. On dispose de n urnes, numérotées de 1 à n . Pour chaque $k \in \llbracket 1, n \rrbracket$, l'urne k est composée de k boules numérotées de 1 à k . On choisit une urne au hasard de manière équiprobable puis on tire une boule au hasard dans cette urne, toujours de manière équiprobable. On note

- X_n la v.a.r. égale au numéro de l'urne choisie
- Y_n la v.a.r. égale au numéro de la boule tirée

Compléter la fonction suivante pour qu'elle renvoie une simulation du couple (X_n, Y_n) :

```

1 def simulXY(n):
2     X = rd.randint(1,n+1)
3     Y = rd.randint(1,X+1)
4     return [X,Y]
```

Exercice 2 : On considère deux pièces. La première est équilibrée et la seconde est truquée, de telle sorte qu'elle tombe sur **Pile** avec probabilité $p \in]0, 1[$. On considère l'expérience suivante, décrite en plusieurs étapes :

1. On lance la première pièce jusqu'à ce qu'elle tombe sur **Pile**.
2. Si n désigne le nombre de lancers effectués à l'étape 1, on remplit une urne avec n boules, numérotées de 1 à n .
3. On effectue un unique tirage d'une boule dans l'urne remplie à l'étape 2.
4. Si k désigne le numéro de la boule tirée à l'étape 3, on effectue une suite de k lancers de la deuxième pièce.

On note :

- X la variable aléatoire égale au nombre de lancers effectués à l'étape 1.
- Y la variable aléatoire égale au numéro de la boule tirée à l'étape 3.
- Z la variable aléatoire égale au nombre de **Face** obtenus à l'étape 4.

```

1 def simulXYZ(p):
2     X = rd.geometric(1/2)
3     Y = rd.randint(1, X+1)
4     Z = rd.binomial(Y, 1-p)
5     return [X,Y,Z]
```

Bonus : proposer une fonction **Python** similaire à la précédente qui simule X , Y et Z , mais avec un protocole légèrement modifié. Cette fois-ci, la pièce lancée à l'étape 1 est choisie au hasard entre les deux pièces, et ce de manière équiprobable. A l'étape 4, on lance la pièce qui n'a pas été lancée à l'étape 1.

```

1 def simulXYZbis(p):
2     if rd.random() < 1/2:
3         a = 1/2
4         b = p
5     else:
6         a = p
7         b = 1/2
8     # a est la probabilité que la première pièce lancée tombe sur Pile
9     # b est la probabilité que la deuxième pièce lancée tombe sur Pile
10    X = rd.geometric(a)
11    Y = rd.randint(1, X+1)
12    Z = rd.binomial(Y, 1-b)
13    return [X,Y,Z]
```