Exercice 1 :[Question barrière] Soit $n \geq 2$ un entier fixé. On dispose de n urnes, numérotées de 1 à n. Pour chaque $k \in [\![1, n]\!]$, 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) :

```
def simulXY(n):
    ______________________________
    ______________________________
    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.

```
def simulXYZ(p):
    X = _________________________
    Y = __________________________
    Z = ___________________________
    return [X,Y,Z]
```

<u>Bonus</u> : 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.