

Exercice 1 : [Question barrière] On lance indéfiniment une pièce équilibrée. On note alors X la variable aléatoire égale au rang du lancer où, pour la première fois, on obtient deux *Pile* consécutifs. Si on n'obtient jamais deux *Pile* consécutifs, on conviendra que X vaut -1 .

Par exemple, si on obtient dans cet ordre : *Pile, Face, Face, Pile, Pile, Pile, Face,...* alors X prend la valeur 5.

Recopier et compléter la fonction **Python** ci-dessous afin qu'elle simule les lancers de la pièce jusqu'à l'obtention de deux *Pile* consécutifs, et qu'elle renvoie le nombre de lancers effectués, c'est-à-dire la valeur que prend X .

```
1 def simulX():
2     tirs = 0
3     pile = 0
4     while pile < 2 :
5         if rd.random() < 1/2:
6             pile = pile + 1
7         else:
8             pile = 0
9             tirs = tirs + 1
10    return tirs
```

Exercice 2 : On considère une urne contenant 1 boule rouge et 1 boule verte. On dispose par ailleurs d'une réserve infinie de boules rouges et vertes. On effectue une série infinie de tirages dans cette urne en suivant le protocole suivant :

- si on tire une boule rouge, alors on la remet dans l'urne et on rajoute deux nouvelles boules rouges dans l'urne,
- si on tire une boule verte, alors on la remet dans l'urne et on rajoute une nouvelle boule rouge ainsi que deux nouvelles boules vertes dans l'urne.

On note, pour tout entier $n \geq 2$, Y_n la variable aléatoire égale au rang du tirage à l'issue duquel, pour la première fois, le nombre de boules vertes est supérieur ou égal à n . Compléter la fonction **Python** suivante pour qu'elle simule Y_n :

```
1 def simulY(n):
2     r = 1
3     v = 1
4     Y = 0
5     while v < n:
6         numero_boule = rd.randint(1, r + v + 1)
7         if numero_boule <= r:
8             r = r + 2
9         else:
10            r = r + 1
11            v = v + 2
12            Y = Y + 1
13    return Y
```

Nous aurions pu simuler différemment la variable aléatoire Y_n :

```
1  def simuly(n):
2      r = 1
3      v = 1
4      Y = 0
5      while v < n:
6          if rd.random() <= r / (r+v):
7              r = r + 2
8          else:
9              r = r + 1
10             v = v + 2
11             Y = Y + 1
12     return Y
```

On peut donc retenir que l'événement « On tire une boule rouge » peut se simuler de deux manières différentes :

- `if rd.random() <= r / (r+v):`
- `if rd.randint(1,r+v+1) <= r:`

où r et v désignent respectivement le nombre de boules rouges et le nombre de boules vertes dans l'urne.