

1. SIMULER LE LANCER D'UN DÉ

On souhaite créer un programme qui simule le lancer d'un dé cubique équilibré.

Voici un algorithme qui répond à l'objectif fixé, et le programme correspondant, dans le langage Python :

Définition : fonction `lancer_de()`
renvoyer un entier aléatoire entre 1 et 6

```
from random import randint
def lancer_de():
    return randint(1,6)
```

Saisir ce programme Python et simuler plusieurs lancers de dés : pour cela, on appelle la fonction `lancer_de()`, dont le résultat est aléatoire :

```
>>> lancer_de( )
>>> 2
```

2. LES PETITS CHEVAUX

Au jeu des petits chevaux, un joueur ne peut engager un cheval dans la partie que s'il obtient un six avec le dé. Thomas débute une partie. Il constate qu'il doit faire cinq lancers pour obtenir son premier six et débiter la partie. Il déclare : « Je n'ai vraiment pas de chance aujourd'hui ! ».

Pour déterminer si Thomas a un ressenti justifié, on va déterminer combien de coups, en moyenne, un joueur doit jouer pour engager un cheval lorsqu'on simule un grand nombre de parties.

- (a) Voici un algorithme incomplet qui simule le début d'une partie et renvoie le nombre de lancers de dé consécutifs jusqu'au premier six. Compléter les trois dernières lignes de cet algorithme et du programme correspondant en Python. *On pourra faire appel à toute fonction préalablement définie.*

Définition : fonction `rang_premier_six()`
 $\text{dé} \leftarrow \text{lancer_de}()$
 $\text{compteur} \leftarrow 1$
 Tant que dé est différent de 6
 $\text{dé} \leftarrow \dots$
 $\text{compteur} \leftarrow \dots$
 renvoyer ...

```
def rang_premier_six():
    de = lancer_de()
    compteur = 1
    while de != 6 :
        de = ...
        compteur = ...
    return ...
```

- (b) On note `nb_parties` un nombre entier représentant un nombre de parties jouées au jeu des petits chevaux. On simule `nb_parties` débuts de partie et on détermine le nombre moyen de coups nécessaires pour engager un cheval lors de la réalisation de cet échantillon. Compléter les deux dernières lignes de cet algorithme et du programme correspondant en Python.

Définition : fonction `rang_moyen(nb_parties)`
 $\text{somme} \leftarrow 0$
 Pour i allant de 1 à nb
 $\text{somme} \leftarrow \dots$
 renvoyer ...

```
def rang_moyen(nb_parties):
    somme = 0
    for i in range(nb_parties):
        somme = ...
    return ...
```

Exécuter le programme Python pour de grandes valeurs de `nb_parties`. Que constate-t-on ?

- (c) Après avoir effectué ces simulations, que peut-on dire à Thomas au sujet de son ressenti ?
 (d) Thomas, qui joue très souvent, affirme qu'il a constaté que dans plus de la moitié des cas, les joueurs ont besoin d'au plus 4 lancers pour faire un 6. A-t-il raison ?

3. LE SIX EN SIX COUPS

Le jeu du *Six en six coups* consiste à lancer au plus six fois un dé cubique équilibré. Dès que le joueur obtient un six, il est gagnant. Si aucun six n'apparaît au cours des six lancers, le joueur est perdant.

On souhaite savoir si le jeu est à l'avantage du joueur ou non, c'est-à-dire si la probabilité de gagner est supérieure à 0,5.

- (a) Écrire un algorithme permettant de simuler un nombre variable de parties et de calculer la fréquence des parties gagnantes. Écrire le programme correspondant à cet algorithme en Python.
 (b) Estimer la probabilité de gain à ce jeu à l'aide du programme.
 (c) (Niveau 1ère) Déterminer, sans simulation, la probabilité de gain au jeu du *Six en six coups*.