Le jeu de Monty-Hall

Le jeu oppose un présentateur de télévision à un candidat (le joueur). Le joueur est placé devant trois portes fermées. Derrière l'une d'entre elles, se trouve une voiture et derrière chacune des deux autres se trouve une chèvre.

Le joueur doit tout d'abord désigner une porte.

Puis le présentateur ouvre une porte qui n'est ni celle choisie par le candidat, ni celle cachant la voiture (le présentateur sait quelle est la bonne porte dès le début).

Le candidat a alors le droit d'ouvrir la porte qu'il a choisie initialement, ou d'ouvrir la troisième porte. S'il ouvre la porte derrière laquelle se trouve la voiture, il l'emporte.

On souhaite étudier les deux stratégies suivantes que le joueur peut suivre une fois que le présentateur a ouvert une porte cachant une chèvre :

(a) conserver la porte choisie initialement,

(b) changer de porte.

Quelle stratégie vous parait gagnante ?

Exercice 1 . Simulation en Python Voici l'algorithme permettant de comparer les stratégies **(a)** et **(b)**.
Le tester et préciser quelle est la stratégie gagnante ?

```
from random import *
n=int(input("nombre de tests ?"))
scorea=0#score en conservant la porte du début
scoreb=0#score en changeant la porte du début.
for i in range(n):
    #porte devant la voiture
    v=randint(1,3)
    #porte choisie par le joueur au début
    j=randint(1,3)
    #porte ouverte par le présentateur
    p=1
    while(p==v)or(p==j):#cette boucle sert juste à s'assurer que la
   porte ouverte par me présentateur ne soit ni celle du joueur, ni
   celle de la voiture.
        p=p+1
    if j==v:
        scorea=scorea+1
    else:
        scoreb=scoreb+1
print("sans changer :", scorea,"; en changeant : ", scoreb)
```

Exercice 2 . On nomme C_1, C_2 et V les trois portes selon qu'elles cachent la chèvre 1, 2 ou la voiture.

[1] On considère la stratégie **(a)**, représenter la situation par un arbre pondéré et donner la probabilité de gagner la voiture pour le joueur.

[2] On considère maintenant la stratégie **(b)**. Représenter l'arbre décrivant le déroulement de l'expérience puis déterminer la probabilité de de gagner la voiture pour le joueur.

La boucle for en python

Le vélo-club d'une ville compte aujourd'hui 220 adhérents. Chaque année, il y a 8% des adhérents qui quittent le club et 22 nouveaux adhérents. Le directeur a élaboré ce programme nommé *veloclub.py* en python qui calcule le nombre d'adhérents plusieurs année après.

```
nba=220
n=int(input("Combien d'années de calcul ?"))
for k in range(1,n+1):
    nba=nba*0.92+22
    print("Dans ", k, " années :",nba, " adhérents")
```

`nba=220`	la variable nba (nombre d'adhérents) est initialisée à 220
`n=int(input("Combien d'années de calcul ?"))`	récupération du nombre d'années de calculs demandés
`for k in range(1,n+1):`	instruction de répétition qui va demander de parcourir les instructions dans la boucle pour k prenant toutes les valeurs entières de 1 à n. Le bloc d'instruction est délimité par l'indentation vers la droite.
`nba=nba*0.92+22`	Calcul du nouveau nombre d'adhérents et enregistrement de la nouvelle valeur dans ma variable nba
`print("Dans ", k, " années :",nba, " adhérents")`	Affichage du résultat

Détail de l'exécution pour $n = 4$

Étape	k	nba	Calcul et Affichage
Avant le début de la boucle for		$nba = 220$	
Fin de la 1re itération	1	224,4	On a calculé $nba = 220 * 0.92 + 22$ Affichage "Dans 1 années : 224.4 adhérents"
Fin de la 2e itération	2	228,448	On a calculé $nba = 224.4 * 0.92 + 22$ Affichage "Dans 2 années : 228.448 adhérents"
Fin de la 3e itération	3	232.172	On a calculé $nba = 228.448 * 0.92 + 22$ Affichage "Dans 3 années : 232.17216 adhérents"
Fin de la 4e itération	4	235.598	On a calculé $nba = 232.17216 * 0.92 + 22$ Affichage "Dans 4 années : 235.598 adhérents"

Exercice 3 .

En vous basant sur les exemples ci-dessous, répondre aux questions du bas de la page.

Un programme	Son exécution

```
print("essai 1")
for k in range(4):
    print(k)


print("essai 2")
for k in range(1,4):
    print(k)

print("essai 3")
for k in range(2,7,2):
    print(k)

print("essai 4")
for k in range(22,9,-4):
    print(k)
```

```
essai 1
0
1
2
3
essai 2
1
2
3
essai 3
2
4
6
essai 4
22
18
14
10
```

[1] Lors de l'exécution d'un programme, l'instruction python :

```
for k in range(1,8)
```

Quelles sont les valeurs prises successivement par k ?

[2] Lors de l'exécution d'un programme, l'instruction python :

```
for k in range(0,7)
```

Quelles sont les valeurs prises successivement par k ?

[3] Lors de l'exécution d'un programme, l'instruction python :

```
for k in range(6)
```

Quelles sont les valeurs prises successivement par k ?

[4] Pour que k prenne les valeurs entières entre 1 et 42, quelle instruction peut-on utiliser ?

Exercice 4 . Le programme s'appelle *TP2_exercice3*

On considère l'algorithme ci-dessous, compléter le tableau décrivant son utilisation.

```
x=15
for i in range(1,6):
  x=2*x-10
  print("pour i =", i, "; x=", x)
```

Étape	i	x	affichage
Avant le début de la boucle for		15	
Fin de la 1re itération	1		
Fin de la 2e itération	2		
Fin de la 3e itération			
Fin de la 4e itération			
Fin de la 5e itération			
Fin de la 6e itération			

Exercice 5 . Enregistrer le programme sous le nom *TP2_exercice4*

Farah et Sean ont racheté la start-up biotech Hera en 2022. Cette année-là, l'entreprise réalise 4 millions d'euros de chiffre d'affaires (CA). Farah et Sean estiment que le CA devrait augmenter de 5% par an.

[1] Calculer le CA prévu au bout de 1 an :

[2] Calculer le CA prévu au bout de 2 an :

[3] Compléter le programme ci-dessous pour qu'il calcule le CA prévu pour les 10 années à venir :

```
CA = ..................
for k in ........................... :
    CA = ....................
    print("année :", ............. ," CA = ", ................)
```

Exercice 6 . Enregistrer le programme sous le nom *TP2_exercice5*

Chaque année, entre Août et Octobre, plusieurs milliers de papillons monarques quittent les États-unis et le Canada pour migrer vers le Sud en Californie et au Mexique. Alors qu'on en dénombrait 4,5 millions en 1980, ces insectes n'étaient plus que 19 milliers en 2020. On suppose que, dans le années à venir, la diminution constatée de 12% par an va continuer.

Ecrire un programme en Python pour estimer la population de monarques pour les 20 année suivantes.