

I– SIMULATION D'EXPÉRIENCES ALÉATOIRES (RAPPEL TD01)



1. Utilisation de rand()

rand()
l'instruction **rand()** permet d'obtenir un nombre décimal pseudo-aléatoire dans l'intervalle $]0; 1[$.

Exercice 1

1. Créer un code permettant successivement 10 appels de la fonction **rand()**.
2. Comparer le résultat avec celui obtenu par d'autres élèves de la classe.

Exercice 2

On considère deux dés équilibrés à 6 faces. Que fait le code ci-dessous ?

```
function y = f(v)
n = 100
c = 0
j = 0
while j < n
    x = 1 + floor(6*rand())
    y = 1 + floor(6*rand())
    if x + y == v then
        c = c + 1
    end
    j = j + 1
end

y = c/n
endfunction
```

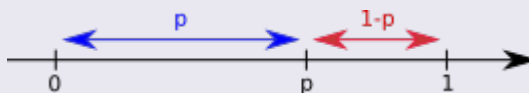
Exercice 3

Écrire un code où l'utilisateur doit deviner un nombre entre 1 et 100 choisi aléatoirement par l'ordinateur. L'utilisateur devra être guidé après chaque proposition (« *trop grand* » ou « *trop petit* »).

2. Simulation d'une expérience de Bernoulli de paramètre p

Méthode

Une variable aléatoire suivant une loi de probabilité de Bernoulli de paramètre p peut être simulée par un test d'appartenance d'une valeur **rand()** à l'intervalle $[0; p]$ (succès) ou $[p; 1]$ (échec).



Exercice 4

On considère une urne comportant 3 boules blanches et 5 boules noires.

1. Créer une fonction **tirage()** renvoyant le mot « *blanche* » ou « *noire* ».
2. Simuler 100 000 fois cette expérience aléatoire et afficher la fréquence d'apparition des boules blanches.

Exercice 5 *extrait d'Ecricome 2019*

L'exercice fait intervenir une suite (c_n) dont le terme général est : $c_n = 1 - \frac{2^n - 1}{3^{n-1}}$.

On considère le script Scilab suivant :

```
n = 1
c = 1-(2^n-1)/3^(n-1)
while c < 0.95
    n = n + 1
    c = 1-(2^n-1)/3^(n-1)
end
disp(n)
```

Après exécution on obtient l'affichage suivant : 11. Interpréter le résultat dans le contexte de l'énoncé.

Exercice 6 *extrait de BCE BSB 2019*

On considère les suites $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}}$ définies par $a_0 = 2$, $b_0 = 0$ et pour tout $n \in \mathbb{N}$:

$$a_{n+1} = 2a_n + 3^n \text{ et } b_{n+1} = 3b_n + 3^n$$

Quelle instruction faut-il ajouter en ligne 4 dans le programme suivant pour qu'il affiche la valeur de a_n , l'entier n étant donné par l'utilisateur (on justifiera la réponse) ?

1. $a = 2 * a + 3 \wedge n$
2. $a = 2 * a + 3 \wedge i - 1$
3. une autre instruction à préciser.

```
1 n = input("n?")
2 a = 2
3 for i=1:n
4     ...
5 end
6 disp(a)
```

Exercice 7 *extrait de BCE ESCP 2019*

On considère les deux suites réelles $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ définies par

$$u_0 = 2, b_0 = 0 \text{ et pour tout } n \in \mathbb{N}, \begin{cases} u_{n+1} = \frac{1}{2}(u_n + v_n) \\ v_{n+1} = \frac{1}{2}(u_{n+1} + v_n) \end{cases}$$

Compléter le script Scilab suivant qui permet de déterminer u_n et v_n pour une valeur de n entrée par l'utilisateur.

```
n = input("entrez la valeur de n :")
u = .....
v = .....
for k=1:n
    u = .....
    v = .....
end
disp(u)
disp(v)
```

Exercice 8 *extrait de BCE ESCP 2018*

On considère $I_0 = \int_1^e t \, dt$ et pour tout $n \in \mathbb{N}^*$, $I_n = \int_1^e t(\ln t)^n \, dt$

1. Calculer I_0
2. Montrer à l'aide d'une intégration par parties que

$$2I_{n+1} + (n+1)I_n = e^2 \quad (*)$$

3. Utiliser la relation $(*)$ pour compléter le script Scilab suivant afin qu'il calcule et affiche I_n pour une valeur de n entrée par l'utilisateur.

```
n = input("entrez la valeur de n :")
I = .....
for k=1:n
    I = .....
end
disp(I)
```