

Correction TP 6 : Aléatoire

EXERCICE 1. Commençons par découvrir la commande qui sera au centre de ce TP.

1. Entrer la commande `rand()` dans la console Scilab, appuyez sur entrée, recommencer 10 fois, qu'obtenez vous ?

Cela dépendra de vous, mais vous trouverez dix nombres aléatoires entre 0 et 1.

2. En vous aidant du titre du TP, que fait la fonction `rand` ?

La fonction `rand` envoie un nombre aléatoire, ou plutôt pseudo aléatoire, entre 0 et 1.

En fait, le réel obtenu n'est pas vraiment aléatoire. En effet, l'ordinateur possède une liste gigantesque de nombres et, lorsque vous utilisez la fonction `rand` il vous en envoie un morceau. Cette liste est suffisamment grande pour que le nombre obtenu semble aléatoire, mais dans le cas de très longs algorithmes, cela peut être problématique. Il existe cependant de vrais générateurs de nombre aléatoire, fonctionnant avec des phénomènes purement quantiques.

3. Écrire le programme suivant sous Scilab :

```
n=10000
for i=1 :n
    if rand()>1 | rand()<0 then
        disp("Lever la main pour appeler le professeur parce que manifestement j'ai mal recopié le programme")
    end
end
end
```

Que pouvez vous en déduire sur les valeurs que peut prendre la fonction `rand()` ?

Vous obtenez donc un nombre entre 0 et 1.

4. Tapez maintenant les commandes suivantes :

```
rand(2,2)
```

```
rand(2,3)
```

```
rand(3,2)
```

Qu'obtenez vous ? Qu'en déduisez vous sur le résultat de la commande `rand(n,m)` ?

Vous obtenez une matrice de taille $n \times m$ avec des coefficients aléatoires.

EXERCICE 2. Commençons à simuler des phénomènes aléatoires.

1. Taper le programme suivant et expliquer avec des mots ce qu'il fait.

```
if rand()<1/3 then
    disp(0)
else
    disp(1)
end
```

On tire un nombre aléatoire, si celui ci est plus petit que $\frac{1}{3}$, le programme renvoie 0, sinon, il renvoie 1.

2. Modifier le programme précédent pour simuler un lancer de pièce avec une pièce équilibrée. On doit donc obtenir soit 'Pile', soit 'Face' avec une probabilité $\frac{1}{2}$.

```
if rand()<1/2 then
    disp("Pile")
else
    disp("Face")
end
```

3. Modifier le programme précédent pour simuler un tirage dans une urne constitué de 3 boules rouges et 2 boules vertes. On doit donc obtenir soit 'rouge', soit 'verte' avec une probabilité que je vous laisse calculer.

```
if rand()<3/5 then
    disp("Rouge")
else
    disp("Verte")
end
```

EXERCICE 3. Cherchons maintenant à simuler des événements possédant plus de deux résultats possibles.

1. En testant avec Scilab les commandes proposées, compléter les phrases suivantes :

La commande `2×rand()` renvoie un nombre pseudo-aléatoire compris entre : 0 et 2.

La commande `rand()+1` renvoie un nombre pseudo-aléatoire compris entre : 1 et 2.

La commande `6×rand()+1` renvoie un nombre pseudo-aléatoire compris entre : 1 et 7.

2. Rappeler la fonction de la commande `floor`, si besoin utiliser l'aide de Scilab

Elle revoie la partie entière de l'argument x

3. En utilisant les commandes `floor` et `rand()`, écrire une commande Scilab donnant un nombre entier pseudo aléatoire entre 1 et 6 choisi de façon équiprobable.

```
floor(6×rand()+1)
```

4. En déduire un programme Scilab permettant de simuler le lancer d'un dé non biaisé.

```
floor(6×rand()+1)
```

5. Rédiger un programme pour simuler un tirage dans une urne constitué de 3 boules rouges, 2 boules vertes et 6 boules bleues. On doit donc obtenir soit 'rouge', soit 'verte', soit 'bleue' avec une probabilité que je vous laisse calculer.

```
alea=rand()
if alea<3/11 then
    disp("Rouge")
elif alea<5/11
    disp("Verte")
else
    disp("Bleue")
end
```

EXERCICE 4 (Adapté de EDHEC 2005).

Un mobile se déplace sur les points à coordonnées entières d'un axe d'origine 0. À l'instant 1, le mobile est à l'origine. Le mobile se déplace selon la règle suivante : s'il est sur le point d'abscisse k à l'instant n , alors à l'instant $(n + 1)$, il sera sur le point d'abscisse $(k + 1)$ avec la probabilité $\frac{2}{3}$ ou le point d'abscisse 0 avec la probabilité $\frac{1}{3}$. Pour tout $n \in \mathbb{N}$, on note X_n l'abscisse de ce point à l'instant n , on a donc $X_1 = 0$.

Rédiger un programme Scilab qui demande un entier n à l'utilisateur et simule cette expérience aléatoire en renvoyant le vecteur de taille n des positions X_1, X_2, \dots, X_n .

Je vous laisse réfléchir à cet exercice, en cas de besoin, vous trouverez un corrigé de l'épreuve EDHEC 2005 sur internet sans grande difficulté.