



- 1°) Générer un nombre aléatoire dans l'intervalle $[0; 1[$.
- 2°) Simuler le lancer d'un dé.
- 3°) a) Simuler 20 lancer d'un dé.
b) Déterminer le nombre de fois où la face 6 a été obtenue.
c) Représenter les résultats obtenus à ces 20 lancers à l'aide d'un diagramme en bâtons.



! Les résultats numériques obtenus sur votre calculatrice peuvent être différents de ceux affichés sur cette fiche.

Génération d'un nombre "aléatoire" dans l'intervalle $[0 ; 1[$

Utiliser l'instruction **rand** :

- Touche **MATH** déplacer le curseur sur l'option **PRB**
- Choisir **1: rand** et appuyer sur **ENTER**.
- Appuyer plusieurs fois sur **ENTER** permet d'obtenir plusieurs simulations.

```
MATH NUM CPX PRB
1:rand
2:nPr
3:nCr
4:!
5:randInt(
6:randNorm(
7:randBin(
```

```
rand
.9435974025
```

Génération d'un nombre "aléatoire" entier compris entre deux bornes

Utiliser l'instruction **randint** :

- Touche **MATH** option **PRB** menu **5: randint(**
- Préciser les bornes séparées par une virgule.
- Fermer la parenthèse et appuyer sur **ENTER**.

Par exemple, l'instruction **randint(1,6)** génère un nombre aléatoire entier compris entre 1 et 6 et peut donc être utilisée pour simuler le lancer d'un dé.

```
MATH NUM CPX PRB
1:rand
2:nPr
3:nCr
4:!
5:randInt(
6:randNorm(
7:randBin(
```

```
randint(1,6) 4
```

Génération de plusieurs nombres "aléatoires" entiers compris entre deux bornes

Pour générer plusieurs nombres aléatoires :

L'instruction **randint(1,6,20)** génère 20 nombres aléatoires entier compris entre 1 et 6.

→ Utiliser les flèches pour faire défiler les résultats.

→ L'instruction **randint** s'utilise de la manière suivante :

`randint(borne inf. , borne sup. , nombre d'essais)`

Pour compter le nombre de 6 obtenus :

Stocker les résultats dans une liste.

- Touche **STO>** puis **L1** (touches **2ND 1**)

Puis trier la liste.

- Menu **LIST** (touches **2ND STAT**)
- Option **OPS** menu **1: SortA(**

On peut alors afficher la liste triée :

- **L1** (touches **2ND 1**)

→ Sur l'exemple ci-contre, la face 6 a été obtenue 2

```
randint(1,6,20)
(5 6 3 1 3 4 3 ...
```

```
randint(1,6,20)
... 2 2 5 3 2 5 6)
```

```
randint(1,6,20)
... 2 2 5 3 2 5 6)
Ans→L1
```

```
randint(1,6,20)
... 2 2 5 3 2 5 6)
Ans→L1
(5 6 3 1 3 4 3 ...
```

```
NAMES OPS MATH
1:SortA(
2:SortD(
3:dim(
4:Fill(
5:seq(
6:cumSum(
7:List(
```

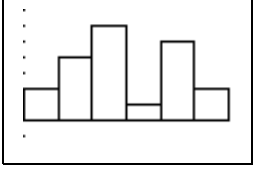
```
SortA(L1) Done
```

```
SortA(L1) Done
L1
(1 1 2 2 2 2 3 ...
```

```
SortA(L1) Done
L1
... 5 5 5 5 6 6)
```

fois.	
-------	--

Représentation graphique des résultats

<p>Si les résultats sont stockés dans la liste 1 :</p> <ul style="list-style-type: none"> - Menu STATPLOT (touches 2ND Y=), puis - 1:Plot1 ENTER et régler comme ci-contre : <p>Régler la fenêtre graphique :</p> <ul style="list-style-type: none"> - Instruction ZoomStat (touche ZOOM 9:ZoomStat). <p>Puis touche GRAPH pour visualiser le graphique.</p>	<div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%; border: 1px solid black; padding: 2px; margin-bottom: 2px;"> <pre> STAT PLOTS 1:Plot1...Off ▽ L1 1 2:Plot2...Off ▽ L1 L2 3:Plot3...Off ▽ L1 L2 4:PlotsOff </pre> </div> <div style="width: 50%; border: 1px solid black; padding: 2px; margin-bottom: 2px;"> <pre> ZOOM Plot2 Plot3 Off Off Type: L1 Xlist:L1 Freq:1 </pre> </div> <div style="width: 50%; border: 1px solid black; padding: 2px; margin-bottom: 2px;"> <pre> MEMORY 4:ZDecimal 5:ZSquare 6:ZStandard 7:ZTrig 8:ZInteger 9:ZoomStat 0:ZoomFit </pre> </div> <div style="width: 50%; border: 1px solid black; padding: 2px;">  </div> </div>
--	--

⇒ **Compléments**

Simulation du lancer d'une pièce

<p>L'instruction randInt(0,1) génère un nombre aléatoire entier qui vaut soit 0 soit 1 et peut donc être utilisée pour simuler le lancer d'une pièce.</p> <p>On peut par exemple décider que l'obtention du chiffre 0 correspond à l'apparition de "Pile" et que l'obtention du chiffre 1 correspond à l'apparition de "Face".</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <pre> randInt(0,1) 1 0 0 1 1 0 </pre> </div>
---	---

Autre méthode pour simuler : Utilisation d'une suite de nombres au hasard

Comme la fonction **random** de la calculatrice (instruction **rand**) fourni un nombre aléatoire dans l'intervalle [0 ; 1], la partie décimale de ce nombre peut être considérée comme une suite de dix chiffres au hasard. Ces chiffres peuvent être utilisés pour une simulation.

<p>Simulation du lancer d'une pièce</p> <p>On peut convenir que les chiffres pairs (0, 2, 4, 6, 8) correspondent à l'apparition de "Pile" et que les chiffres impairs (1, 3, 5, 7, 9) correspondent à l'apparition de "Face".</p> <p>L'exemple ci-contre correspond au tirage "P-F-F-P-P-P-P-F-P-P".</p> <p>Simulation du lancer d'un dé</p> <p>On peut convenir de conserver les chiffres correspondant à une face d'un dé (1, 2, 3, 4, 5, 6) et de supprimer les autres chiffres (0, 7, 8, 9).</p> <p>L'exemple ci-contre correspond au tirage "4-1-6-6-5-4-4"</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre> rand .4918686544 </pre> </div> <div style="border: 1px solid black; padding: 5px;"> <pre> rand .4918686544 </pre> </div>
--	--

Simulation d'une situation ou il n'y a pas équiprobabilité

<p>L'instruction int(rand+0,25) génère un nombre aléatoire entier qui vaut 0 dans 75 % des cas et 1 dans 25 % des cas.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <pre> int(rand+.25) 0 1 0 0 0 1 </pre> </div>
---	--

⇒ **Commentaires**

! **Prise en compte de la dernière décimale**

La dernière décimale affichée étant une valeur arrondie ; on peut, pour ne pas risquer de nuire à l'équiprobabilité des résultats, ne pas tenir compte de cette décimale.

Sur l'exemple ci-contre, on peut ne conserver que les chiffres 943597402 et ignorer la dernière décimale.

```
rand
.9435974025
```

! **Prise en compte des zéros non significatifs**

Si il y a des zéros en fin de la partie décimale, ceux-ci ne sont pas affichés. Mais ils doivent être pris en compte pour conserver le caractère équiprobable de la simulation.

Sur l'exemple ci-contre, le résultat affiché ne contient que 8 chiffres. Comme les nombres affichés par la calculatrice contiennent 10 chiffres significatifs, le résultat obtenu est en réalité 0,2795177400. Les deux derniers zéros doivent être pris en compte pour la simulation.

```
rand
.27951774
```

! **Choix de la valeur initiale**

A chaque exécution de **rand**, la TI-83 Plus génère la même suite de nombres aléatoires pour une valeur de départ donnée. La valeur de départ de la TI-83 Plus réglée en usine pour **rand** est 0.

Pour générer une suite de nombre aléatoires différente, mémoriser une valeur de départ différente de zéro dans **rand**.

Pour restaurer la valeur de départ configurée en usine, mémoriser 0 dans **rand**, ou réinitialisez les valeurs par défaut (Voir chapitre 18 de la notice).

Ainsi : si les élèves mémorisent la même valeur dans **rand**, ils trouveront tous les mêmes suites de nombres, si ils mémorisent des valeurs différentes dans **rand**, ils trouveront des suites de nombres différentes.

Remarque : La valeur de départ a également une incidence sur l'instruction **randInt**

```
2009→rand
rand
.831591802
```

```
rand
.9435974025
0→rand
rand
.9435974025
```

✂ **Limitations de l'instruction randInt**

L'instruction **randInt** ne fonctionne pas avec des valeurs décimales par contre elle peut être utilisée avec des entiers négatifs.

```
randInt(-1,0)
-0
-1
0
0
-1
```

✂ **Génération d'un nombre « aléatoire » dans l'intervalle [0 ;n[(n entier)**

Par exemple : **rand5** génère un nombre aléatoire supérieur à 0 et inférieur strictement à 5.

```
rand5
.7334391461
```

✂ **Autre instruction pour simuler un nombre "aléatoire" entier compris entre deux bornes**

Par exemple : pour simuler le lancer d'un dé, on peut utiliser l'instruction : **int(6*rand+1)**.

Quelques précisions sur la formule :
Avec l'instruction **rand**, le nombre aléatoire obtenu est tel que : $0 \leq \text{rand} < 1$ soit :
 $0 \leq 6 * \text{rand} < 6$
 $1 \leq 6 * \text{rand} + 1 < 7$

Avec l'instruction **int**, on obtient la partie entière du nombre aléatoire, c'est-à-dire un entier compris entre 1 et 6.

Autre exemple : pour simuler le lancer d'une pièce, on peut utiliser l'instruction : **int(2*rand)**.

```
int(6*rand+1)
3
```

```
int(2*rand)
1
0
1
0
0
0
```