



- 1°) Générer un nombre aléatoire dans l'intervalle [0 ; 1[.
- 2°) Simuler le lancer d'un dé.
- 3°) a) Simuler 20 lancers 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

Générer un nombre "aléatoire" dans l'intervalle [0 ; 1[

- 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
.1018231622
.7916288018
.3688383045
```

Simuler le lancer d'un dé (Générer 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
```

Simuler 20 lancers d'un dé (Générer plusieurs nombres "aléatoires" entiers compris entre deux bornes)

- Pour générer plusieurs nombres aléatoires :
- L'instruction **randInt(1,6,20)** génère 20 nombres aléatoires entiers 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 **2nde 1**). Valider par **ENTER** .
- # Trier la liste.
- Touche **STATS** et choix **2: SortA(**.
 - Saisir **L1**) et valider par **ENTER**.
 - Touche **STAT** et choix **1: Edit...** pour afficher la liste triée.
- Sur l'exemple ci-contre, la face 6 a été obtenue 5 fois.

```
randInt(1,6,20)
6 3 2 3 6 2 3 ...
```

```
randInt(1,6,20)
1 6 2 1 6 3 3
```

```
randInt(1,6,20)
1 6 2 1 6 3 3
Ans→L1
```

```
entAléat(1,6,20)
(1 1 5 1 3 2 6 ...
Rep→L1
(1 1 5 1 3 2 6 ...
```

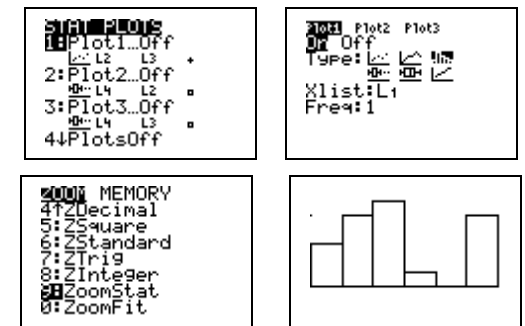
```
2nd) CALC TESTS
1:Edit...
2:SortA(
3:SortD(
4:ClrList
5:SetUPEditor
```

```
SortA(L1) Done
```

L1	L2	L3	1
1			
2			
3			
4			
5			
6			
L1(1)=1			

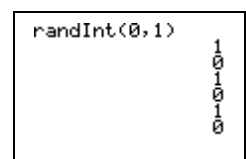
L1	L2	L3	1
1			
2			
3			
4			
5			
6			
L1(20)=6			

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=), - Choix 1:Plot1 ENTER et régler comme ci-contre : <p>Régler la fenêtre graphique :</p> <ul style="list-style-type: none"> - Touche ZOOM choix 9:ZoomStat et ENTER . 	 <p>The screenshots show the following settings:</p> <ul style="list-style-type: none"> StatPlots: Plot1:On, Plot2:Off, Plot3:Off, Type:Normal, Xlist:L1, Freq:1. ZoomStat: 9:ZoomStat selected. Histogram: A bar chart with 5 bars of varying heights.
---	---

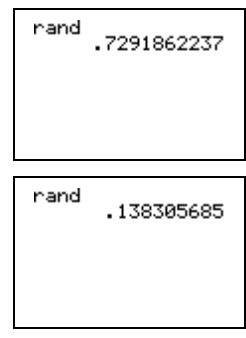
⇒ **Compléments**

Simulation du lancer d'une pièce

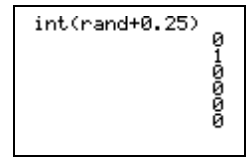
<p>L'instruction randInt(0,1) génère un nombre aléatoire entier qui vaut soit 0 soit 1 et peut donc être utilisée pour simuler le lancer d'une pièce.</p> <p>On peut par exemple décider que l'obtention du chiffre 0 correspond à l'apparition de "Pile" et que l'obtention du chiffre 1 correspond à l'apparition de "Face".</p>	 <p>randInt(0,1)</p> <p>1 0 1 0 1 0</p>
---	--

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

L'instruction **randInt** fournit 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 "F-P-F-F-P-P-P-F-F".</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 "1-3-3-5-6-5"</p>	 <p>rand .7291862237</p> <p>rand .138305685</p>
--	--

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> <p>En effet, on obtient la partie entière d'un nombre aléatoire dans l'intervalle [0,25 ; 1,25[.</p> <p>75% de ces nombres sont dans [0,25 ; 1[et 25% dans [1 ; 1,25[.</p>	 <p>int(rand+0.25)</p> <p>0 1 0 0 0 0</p>
--	--

⇒ **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 284022351 et ignorer la dernière décimale.

```
rand
.2840223519
```

! **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 quatrième résultat affiché ne contient que 9 chiffres. Comme les nombres affichés par la calculatrice contiennent 10 chiffres significatifs, le résultat obtenu pour la simulation est en réalité 0,8938268450.

```
rand
.5186526569
.8124213628
.0912714316
.893826845
.0201154024
.8918302165
```

! **Choix de la valeur initiale**

A chaque exécution de **randInt**, 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 **randInt** est 0.

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

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

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

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

```
2009→rand
rand
.831591802
```

```
0→rand
rand
.9435974025
```

✂ Compléments sur l'instruction rand

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

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

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

Par exemple :

rand 5 génère un nombre aléatoire supérieur à 0 et inférieur strictement à 5.

En fait, la calculatrice multiplie par 5 un nombre aléatoire.

```
rand5
1.696812625
```

✂ 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 :

$$\text{int}(6 * \text{rand} + 1)$$

Quelques précisions sur la formule :

Avec l'instruction **rand**, le nombre aléatoire obtenu est tel que :

$$0 \leq \text{rand} < 1 \text{ donc } 0 \leq 6 * \text{rand} < 6 \text{ donc } 1 \leq 6 * \text{rand} + 1 < 7$$

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 :

$$\text{int}(2 * \text{rand})$$

```
int(6*rand+1)
6
5
6
5
6
```

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