



- 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: NbrAléat** et appuyer sur **entrer**.
- Appuyer plusieurs fois sur **entrer** permet d'obtenir plusieurs simulations.

```
MATH NUM CPX PRB
1: NbrAléat
2: Arrangement
3: Combinaison
4: !
5: entAléat(
6: normAléat(
7: BinAléat(
```

```
NbrAléat
.908318861
```

Simuler le lancer d'un dé (Générer un nombre "aléatoire" entier compris entre deux bornes)

- Utiliser l'instruction **entAléat** :
- Touche **math** option **PRB** menu **5: entAléat(**
 - Préciser les bornes séparées par une virgule.
 - Fermer la parenthèse et appuyer sur **entrer**.
- Par exemple, l'instruction **entAléat(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: NbrAléat
2: Arrangement
3: Combinaison
4: !
5: entAléat(
6: normAléat(
7: BinAléat(
```

```
entAléat(1,6)
1
```

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

- Pour générer plusieurs nombres aléatoires :
- L'instruction **entAléat(1,6,20)** génère 20 nombres aléatoires entiers compris entre 1 et 6.
- Utiliser les flèches pour faire défiler les résultats.
- L'instruction **entAléat** s'utilise de la manière suivante :
- entAléat(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**)
- Puis trier la liste.
- Touche **stats** et choix **2: TriCroi(**
 - Saisir **L1** et valider par **entrer**
 - Touche **stats**) et choix **1: Edite...** pour afficher la liste triée
- Sur l'exemple ci-contre, la face 6 a été obtenue 5 fois.

```
entAléat(1,6,20)
{1 1 5 1 3 2 6 ...
```

```
entAléat(1,6,20)
... 1 1 4 6 6 2 2)
```

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

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

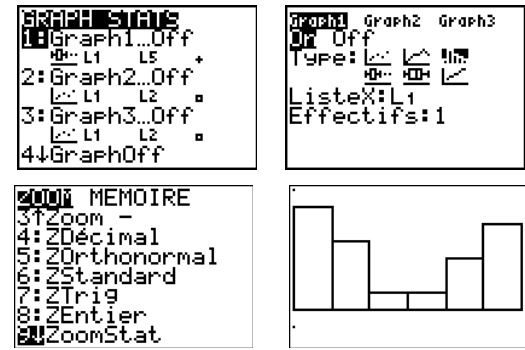
```
CALC TESTS
1: Edite...
2: TriCroi(
3: TriDécroi(
4: EffListe
5: ListesDéfaut
```

```
TriCroi(L1)
Fait
```

L1	L2	L3	1
1	---	---	
1	---	---	
1	---	---	
1	---	---	
1	---	---	
2	---	---	
L1(1)=1			

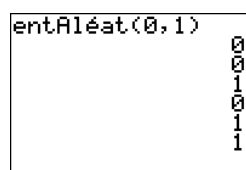
L1	L2	L3	1
1	---	---	
1	---	---	
1	---	---	
1	---	---	
1	---	---	
2	---	---	
L1(20)=6			

Représentation graphique des résultats

<p>Si les résultats sont stockés dans la liste 1 :</p> <ul style="list-style-type: none"> - Menu graph stats (touches 2nde f(x) =), - Choix 1:Graph1 entrer et régler comme ci-contre : <p>Régler la fenêtre graphique :</p> <ul style="list-style-type: none"> - Touche zoom choix 9:ZoomStat et entrer . - Touche graphe pour visualiser le graphique. 	
--	--


⇒ **Compléments**

Simulation du lancer d'une pièce


<p>L'instruction entAléat(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>	
--	---

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

Comme la fonction **Nombre Aléatoire** de la calculatrice (instruction **NbrAléat**) 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.

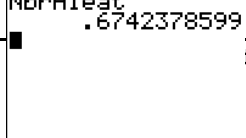
<p>Simulation du lancer d'une pièce</p> <p>On peut convenir que les chiffres pairs (0, 2, 4, 6, 8) correspondent à l'apparition de "Pile" et que les chiffres impairs (1, 3, 5, 7, 9) correspondent à l'apparition de "Face".</p> <p>L'exemple ci-contre correspond au tirage "P-F-F-F-P-P-F-P-F-P".</p> <p>Simulation 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 "1-1-1-4-6-2-4"</p>	
--	---

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

<p>L'instruction ent(NbrAléat +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>Et 75% de ces nombres sont dans $[0,25 ; 1[$ et 25% dans $[1 ; 1,25[$.</p>	
--	---

⇒ **Commentaires**

Prise en compte de la dernière décimale

<p>Statistique et probabilités</p> <p>Simulation</p>	 <p>.s.fr</p>
--	--

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 674237859 et ignorer la dernière décimale.

! 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 8 chiffres. Comme les nombres affichés par la calculatrice contiennent 10 chiffres significatifs, résultat obtenu pour la simulation est en réalité 0,2795177400.

```
NbrAléat
.5833688486
.338072944
.2227754528
.27951774
```

le

! Choix de la valeur initiale

A chaque exécution de **NbrAléat**, la TI-84 Plus génère la même suite de nombres aléatoires pour une valeur de départ donnée.

La valeur de départ de la TI-84 Plus réglée en usine pour **NbrAléat** est 0.

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

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

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

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

```
2009→NbrAléat
NbrAléat 2009
.831591802
```

```
NbrAléat
5041814346
0→NbrAléat 0
NbrAléat
.9435974025
```

☒ Compléments sur l'instruction entAléat

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

```
entAléat(-1,1)
1
-1
0
0
-1
-1
```

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

Par exemple :

NbrAléat 5 génère un nombre aléatoire supérieur à 0 et inférieur strictement à 5. En fait, la calculatrice multiplie par 5 un nombre aléatoire.

```
NbrAléat5
4.486754793
```

☒ 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 :

$$\text{ent}(6 * \text{NbrAléat} + 1).$$

Quelques précisions sur la formule :

Avec l'instruction **NbrAléat**, le nombre aléatoire obtenu est tel que : $0 \leq \text{NbrAléat} < 1$ soit :

$$0 \leq 6 * \text{NbrAléat} < 6$$

$$1 \leq 6 * \text{NbrAléat} + 1 < 7$$

Avec l'instruction **ent**, 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 :

$$\text{ent}(2 * \text{NbrAléat}).$$

```
ent(6*NbrAléat+1)
2
6
2
6
2
6
```

```
ent(2*NbrAléat)
1
1
0
0
0
1
```