



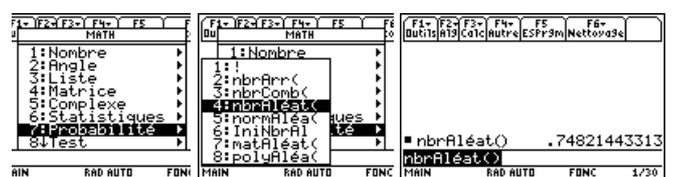
- 1°) Générer un nombre aléatoire dans l'intervalle $[0 ; 1[$.
- 2°) Simuler le lancer d'un dé.
- 3°) a) Simuler 20 lancers d'un dé.
 b) Déterminer le nombre de fois où la face 6 a été obtenue.
 c) Représenter les résultats obtenus à ces 20 lancers à l'aide d'un diagramme en bâtons.



! 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 **HOME** pour ouvrir l'écran de calcul
 - Instruction **MATH** (touches **2ND 5**) puis choix **7:Probabilités** et **▶**
 - Choisir **4: nbrAléat (**, fermer la parenthèse et appuyer sur **ENTER**.
- Appuyer plusieurs fois sur **ENTER** permet d'obtenir plusieurs simulations.



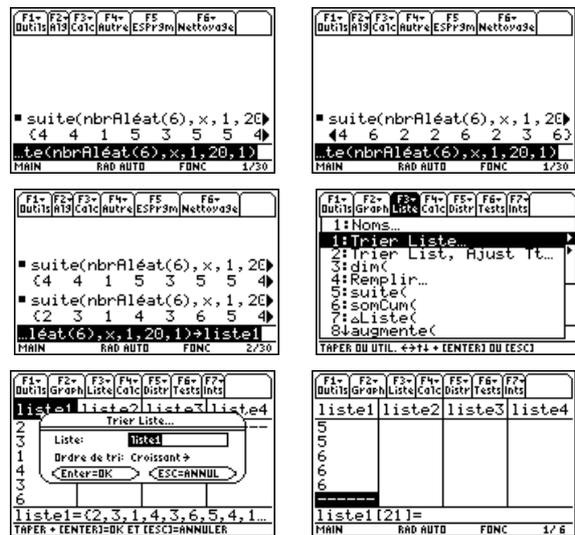
Simuler le lancer d'un dé (Générer un nombre "aléatoire" entier compris entre deux bornes)

- Utiliser comme ci-dessus l'instruction **nbrAléat (** :
- Instruction **MATH** choix 7 puis choix 4
 - saisir le nombre 6, fermer la parenthèse et appuyer sur **ENTER**.
- 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.



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

- Pour générer plusieurs nombres aléatoires, on utilise l'instructions **Suite**. Ici on demande la suite de 20 résultats aléatoires **nbrAléat(6)**. Voir écran ci-contre.
- L'instruction **suite** s'utilise de la manière suivante :
- suite(expression, variable, valeur initiale, valeur finale, pas)
- Utiliser les flèches pour faire défiler les résultats.
- Pour compter le nombre de 6 obtenus :**
- Stocker les résultats dans une liste.
- Touche **STO ▶** puis **liste1** en alphanumérique
 - Puis trier la liste.
 - Touche **APPS** et choix **Stats/Editor**
 - Touche **F3 Liste** choix **2: OPS** puis choix **1: Trier Liste**
- Sur l'exemple ci-contre, la face 6 a été obtenue 3 fois.



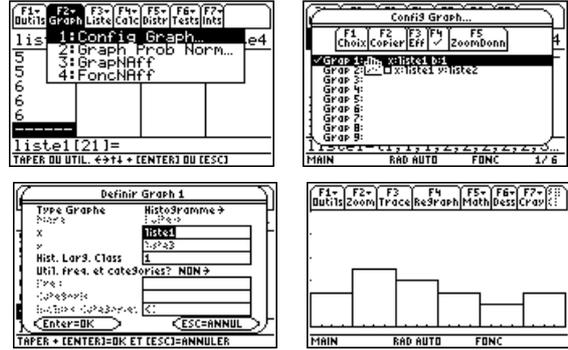
Représentation graphique des résultats

Si les résultats sont stockés dans la liste 1 :

- Touche **F2 Graph Y=** ,
- Choix **1:Config Graph ENTER** sélectionner avec **F1** le graph1 et régler comme ci-contre :

Régler la fenêtre graphique :

- Touche **F5 ZoomDown** pour visualiser le graphique
- ci-contre fenêtre graphique avec $-0,5 < Y < 10$.



⇒ **Compléments**

Simulation du lancer d'une pièce

L'instruction **nbrAléat(1)** génère un nombre aléatoire entier qui vaut soit 1 soit 2 et peut donc être utilisée pour simuler le lancer d'une pièce. On peut par exemple décider que l'obtention du chiffre 1 correspond à l'apparition de "Pile" et que l'obtention du chiffre 2 correspond à l'apparition de "Face".



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.

Simulation du lancer d'une pièce

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". L'exemple ci-contre correspond au tirage "P-P-P-F-F-F-F-P-P-F-F-F".

Simulation du lancer d'un dé

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). L'exemple ci-contre correspond au tirage "4-4-3-1-4-6-2-4"



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

L'instruction **entPréc(nbrAléat() +0,25)** génère un nombre aléatoire entier qui vaut 0 dans 75 % des cas et 1 dans 25 % des cas. En effet, on obtient la partie entière d'un nombre aléatoire dans l'intervalle $[0,25 ; 1,25[$. Et 75% de ces nombres sont dans $[0,25 ; 1[$ et 25% dans $[1 ; 1,25[$.



⇒ **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 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 dernier résultat affiché ne contient que 10 chiffres.

Comme les nombres affichés par la calculatrice contiennent 12 chiffres significatifs, le résultat obtenu pour la simulation est en réalité 0,148232647100.



! **Choix de la valeur initiale**

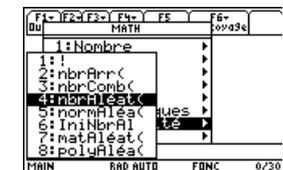
A chaque exécution de **nbrAléat()**, la TI-89 Titanium génère la même suite de nombres aléatoires pour une valeur de départ donnée.

La valeur de départ de la TI-89 Titanium réglée en usine pour **nbrAléat()** est 0.

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

Pour restaurer la valeur de départ configurée en usine, mémoriser 0 dans **InitNbrAl**.

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



✗ Compléments sur l'instruction nbrAléat

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



✗ 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.



✗ 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 :

$$\text{entPréc}(6 * \text{nbrAléat}() + 1).$$

Quelques précisions sur la formule :

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

$$0 \leq 6 * \text{nbrAléat}() < 6$$

$$1 \leq 6 * \text{nbrAléat}() + 1 < 7$$

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 :

$$\text{entPréc}(2 * \text{nbrAléat}()).$$

