

Thème : Dénombrement, Probabilités et Statistiques

Probabilités conditionnelles.

Cet énoncé est celui de la deuxième épreuve orale (épreuve sur dossier) du Capes Externe de mathématiques, proposé aux candidat(e)s le 19 Juillet 2005.

Pour consulter les archives de cette épreuve orale, depuis la session 2005, on se reportera au site officiel du jury, à l'adresse <http://capes-math.org/>

1. L'exercice proposé au candidat

Une usine produit des objets dont un pour cent est défectueux.

On teste ces objets en bout de chaîne.

- Si un objet est défectueux, la probabilité que le test le décèle est égale à 0,9.
- Si un objet n'est pas défectueux, la probabilité que le test le trouve défectueux vaut 0,05.

On teste un objet pris au hasard.

1. Le test donne l'objet comme défectueux, quelle est la probabilité qu'il le soit vraiment ?
2. Les événements « l'objet pris au hasard est défectueux » et « l'objet pris au hasard est donné défectueux par le test » sont-ils indépendants ?

2. Le travail demandé au candidat

En aucun cas, le candidat ne doit rédiger sur sa fiche sa solution de l'exercice. Celle-ci pourra néanmoins lui être demandée partiellement ou en totalité lors de l'entretien avec le jury

Après avoir résolu et analysé l'exercice le candidat rédigera sur sa fiche les réponses aux questions suivantes :

1. Indiquer les classes de lycée dans lesquelles on peut proposer cet exercice ainsi que les notions et outils mis en oeuvre dans sa résolution.
2. Rédiger les questions intermédiaires à proposer pour rendre cet exercice accessible à un élève de lycée.
3. Réaliser un schéma sous forme d'arbre de probabilités pouvant servir de support à la résolution.
4. Proposer un ou plusieurs exercices sur le même thème

Proposition de corrigé avec le Classpad 300

I. L'exercice proposé au candidat

On peut modéliser la situation en considérant que l'univers des possibles est l'ensemble des pièces présentes en fin de chaîne, et qu'une issue élémentaire de l'expérience consiste en le choix d'une quelconque de ces pièces, tous les choix étant équiprobables.

On note D l'événement « la pièce choisie est (réellement) défectueuse », et on désigne par T l'événement « le test trouve que la pièce choisie est défectueuse ».

On sait que $p(D) = 0,01$.

La deuxième hypothèse de l'énoncé exprime que $p_D(T) = 0,9$.

Le troisième hypothèse dit que $p_{\bar{D}}(T) = 0,05$.

La formule des probabilités totales (avec le système complet d'événements D et \bar{D}) donne alors :

$$p(T) = p(D)p_D(T) + p(\bar{D})p_{\bar{D}}(T) = (0,01)(0,9) + (0,99)(0,05) = 0,0585$$

1. La question posée est ici de connaître $p_T(D)$ (probabilité que l'objet soit effectivement défectueux alors qu'il est déclaré tel par le test).

$$\text{On a } p_T(D) = \frac{p(T \cap D)}{p(T)} = \frac{p(D)p_D(T)}{p(T)} = \frac{(0,01)(0,9)}{0,0585} \approx 0,154.$$


2. On demande si les deux événements D et T sont indépendants.

S'ils l'étaient, on aurait $p_T(D) = p(D)$ ce qui n'est manifestement pas le cas.

II. Les calculs de probabilités avec le Classpad

L'exercice proposé en ce 19 juillet 2005 étant très simple, on va compléter ce document par un tour d'horizon de ce que le Classpad est capable de faire en matière de probabilités.

1. Jouons aux dés avec le Classpad

À partir de l'application  on a accès à de nombreuses fonctionnalités, par un simple contact du stylet sur l'icône appropriée (fig1).

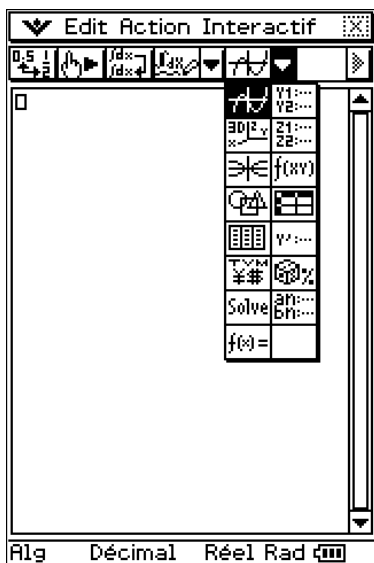


fig1 : à partir de 



fig2 : fenêtre « Probabilités »

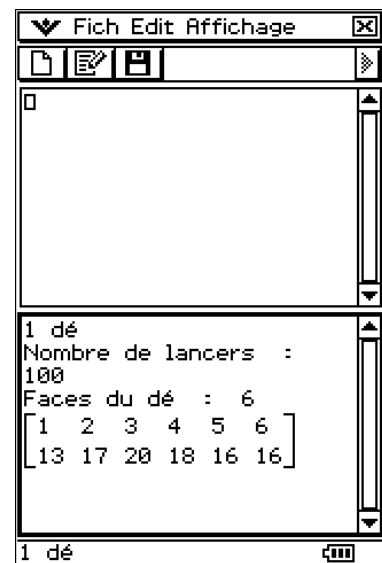



fig3 : 100 lancers (dé à 6 faces)


L'icône  affiche une fenêtre « Probabilités » pour simuler le lancer d'un ou deux dés (fig2).


On voit qu'il est possible de choisir combien de fois on va répéter l'expérience, le nombre de faces (avec deux faces, on lance une pièce, avec par exemple 1 pour pile et 2 pour face).

On peut lancer un seul dé (et observer la fréquence d'apparition des différentes faces), ou deux dés (et observer les valeurs obtenues successivement soit pour la somme soit pour le produit des deux résultats).

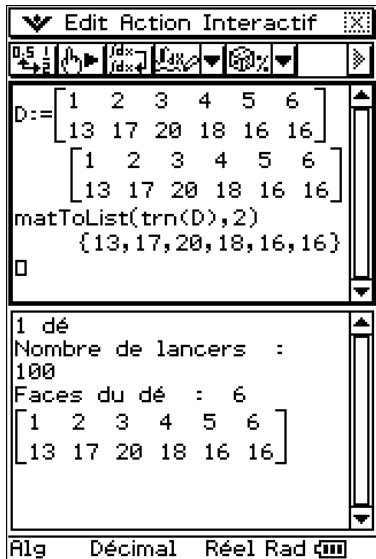
On voit par exemple (fig3) le résultat de 100 lancers d'un dé à six faces. Le résultat est renvoyé dans la moitié inférieure de l'écran. Sous la forme d'une matrice à deux lignes, il indique ici que la face numérotée 4, par exemple, est apparue 18 fois.

Le résultat de cette expérience peut être copié dans la fenêtre supérieure (qui est ici celle de l'application principale). On a ici copié la matrice pour la sauvegarder dans une variable D , et on en a extrait la deuxième ligne, donc les nombres d'apparitions des différentes faces (fig4).

On peut maximiser la fenêtre contenant les résultats, et des appuis sur  permettent de répéter la même expérience (ou d'autres, puisqu'on doit renseigner à chaque fois la fenêtre « Probabilité »). On a ici lancé 100 fois notre dé, trois fois de suite (fig5).

Comme on le voit avec la présence de l'icône , il est possible de sauvegarder les résultats successifs de notre expérience. On peut également supprimer un résultat par « Edit/Supprimer ».

On a ensuite réalisé deux expériences : d'abord lancer 1000 fois une pièce (marquée 1 du côté pile par exemple, et 2 du côté face : on a obtenu 517 fois pile). Puis on a lancé 100 fois deux dés tétraédriques (4 faces) en s'intéressant à la somme des résultats obtenus (fig6).




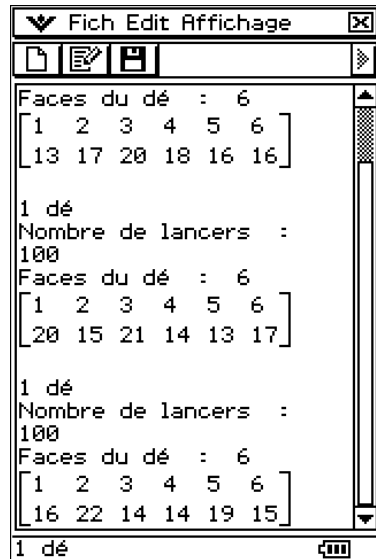
```

Edit Action Interactif
D:= [ 1 2 3 4 5 6 ]
    [13 17 20 18 16 16]
     [ 1 2 3 4 5 6 ]
     [13 17 20 18 16 16]
matToList(trn(D),2)
      {13,17,20,18,16,16}
  
```

```

1 dé
Nombre de lancers :
100
Faces du dé : 6
[ 1 2 3 4 5 6 ]
[13 17 20 18 16 16]
  
```

fig4 : copie dans  Principale



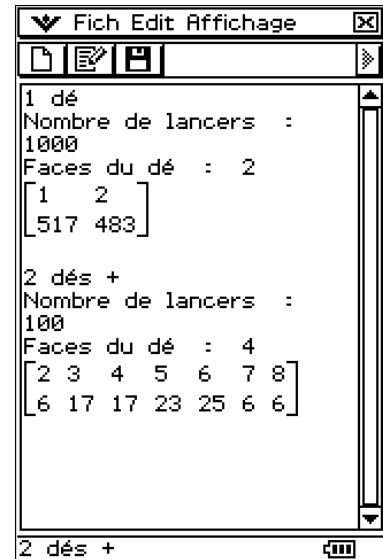
```

Fich Edit Affichage
Faces du dé : 6
[ 1 2 3 4 5 6 ]
[13 17 20 18 16 16]

1 dé
Nombre de lancers :
100
Faces du dé : 6
[ 1 2 3 4 5 6 ]
[20 15 21 14 13 17]

1 dé
Nombre de lancers :
100
Faces du dé : 6
[ 1 2 3 4 5 6 ]
[16 22 14 14 19 15]
  
```

fig5 : répéter l'expérience




```

Fich Edit Affichage
1 dé
Nombre de lancers :
1000
Faces du dé : 2
[ 1 2 ]
[517 483]

2 dés +
Nombre de lancers :
100
Faces du dé : 4
[ 2 3 4 5 6 7 8 ]
[6 17 17 23 25 6 6]
  
```

fig6 : expériences différentes

2. Tirages avec ou sans remise

La même application ouverte par  permet de simuler une expérience consistant à effectuer, dans une urne comptant jusqu'à six objets de types différents (disons des boules de couleurs différentes) un tirage (avec ou sans remise) d'un nombre quelconque de boules et à observer combien on en a obtenu de chaque sorte.

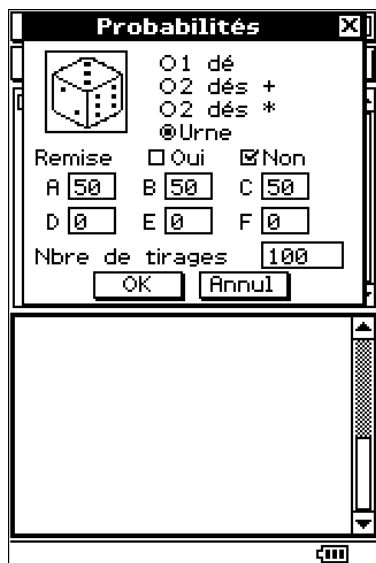


fig7

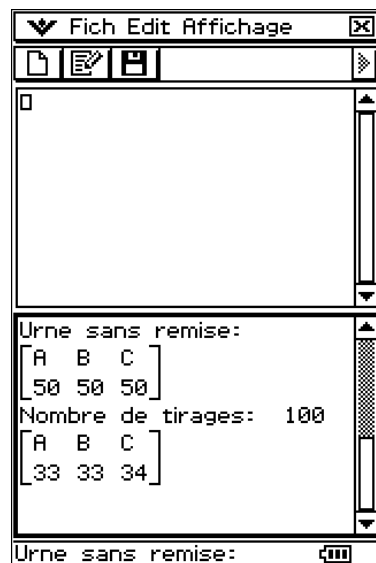


fig8

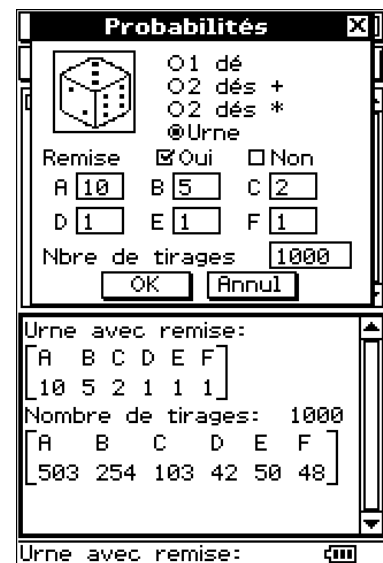


fig9

Exemples de tirages avec ou sans remise

On a ainsi simulé un tirage *sans* remise de 100 boules dans une urne contenant 50 boules de chacun des types A , B et C (fig7), et on en a obtenu respectivement 33, 33 et 34 (fig8). Dans le cas d'un tirage sans remise, le Classpad vérifie naturellement que le nombre de boules tirées est inférieur ou égal au nombre total de boules de l'urne !

On a ensuite simulé un tirage *avec* remise de 1000 boules dans une urne contenant 10 boules de type A , 5 boules de type B , 2 de type C , et une de chacun des types D , E , F . On a obtenu par exemple 503 boules de type A et 48 boules de type F (fig9).

3. Dans l'application « Statistiques »

On ouvre l'application  dans l'écran d'accueil (fig10).

Dans le menu « Calc », on choisit l'option « Distribution » (fig11).

Dans une liste déroulante du demi-écran inférieur, on choisit « DP binomiale ». On peut même cocher une case d'aide pour avoir des indications sur ce que fait la fonction choisie (fig12).

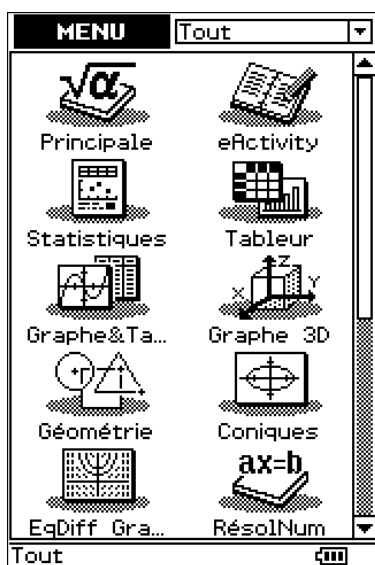
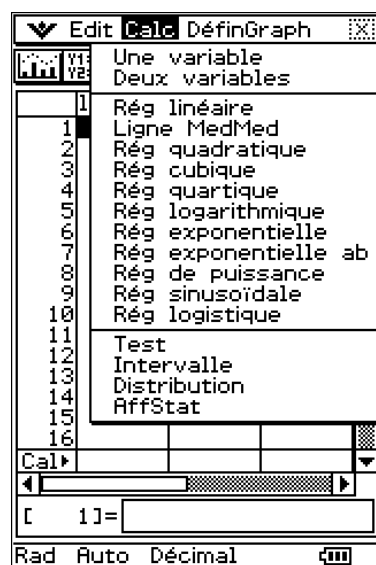
fig10 : l'application 

fig11 : choisir Distribution

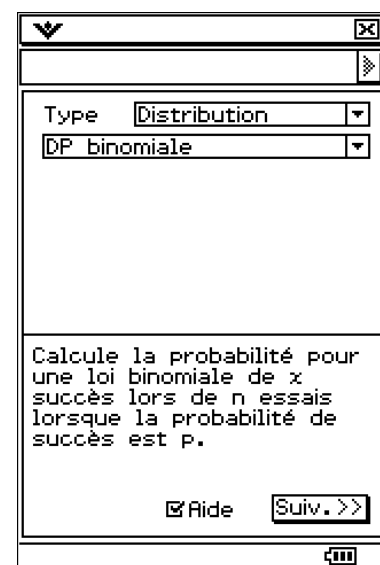


fig12 : loi binomiale

Le choix de **Suiv.>>** affiche une fenêtre permettant de calculer la probabilité qu'on ait x succès dans une succession de n épreuves de Bernoulli indépendantes de même paramètre p (par exemple $x = 8, n = 10$ et $p = 0,75$, cf fig13).

Un nouvel appui sur **Suiv.>>** donne le résultat (ici une probabilité $\approx 0,28$, cf fig14).

Un appui sur l'icône **f** affiche alors l'histogramme de la distribution (fig15).

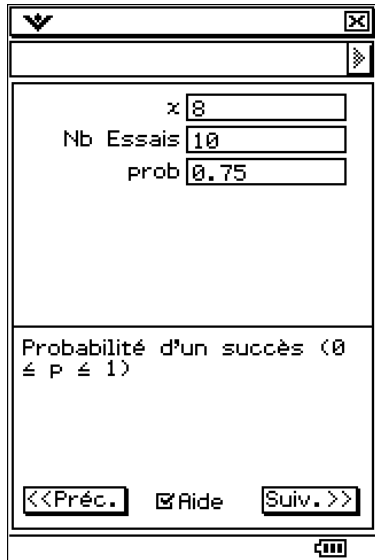


fig13 : les paramètres

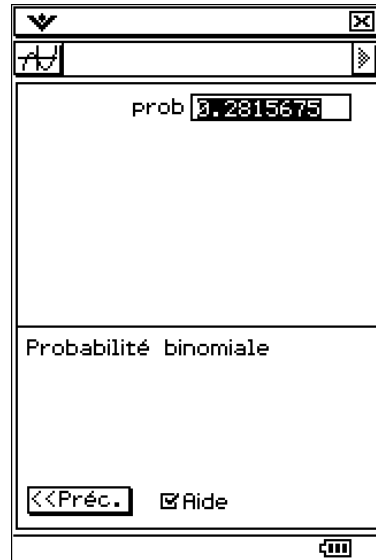


fig14 : la probabilité

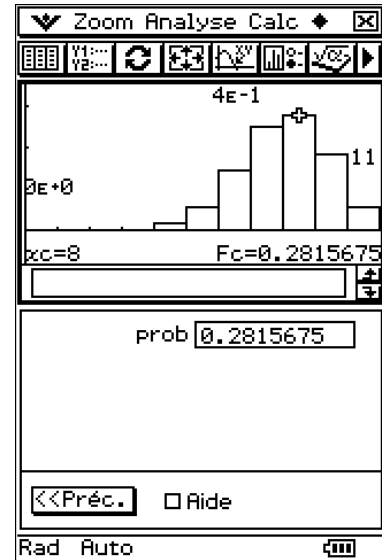


fig15 : l'histogramme

Un appui sur **<<Préc.** permet de modifier les paramètres de la loi binomiale. On a par exemple calculé la probabilité d'obtenir 50 succès dans une loi binomiale de paramètres $n = 100$ et $p = 1/2$, et on a tracé l'histogramme correspondant (fig16).

Cet histogramme est très proche d'une représentation de la densité d'une loi normale qui serait de paramètres $\mu = np = 50$ et d'écart-type $\sigma = \sqrt{npq} = 5$.

Voici d'ailleurs ce que donne l'affichage de cette densité en choisissant au départ la distribution « DP Normale », puis les paramètres $x = 50, \sigma = 5$ et $\mu = 50$ (fig17).

On peut encore étudier de nombreuses lois et des fonctions de répartitions : on voit par exemple la fonction de répartition de la loi binomiale $\mathcal{B}(100, 1/4)$ (et on a calculé la probabilité d'avoir au plus 45 succès, cf fig18).

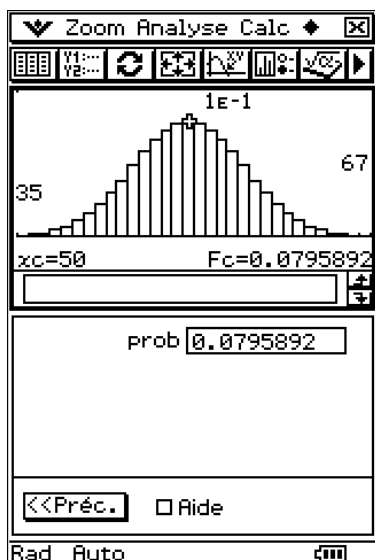


fig16 : la loi $\mathcal{B}(100, 0.5)$

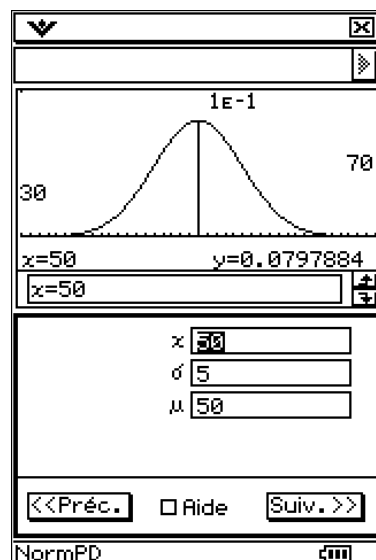


fig17 : loi normale

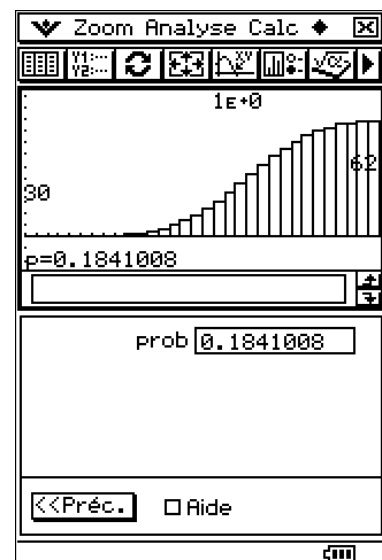




fig18 : foncⁿ de répartition

3. Probabilités dans l'application principale

On peut effectuer toutes sortes de calculs ayant trait aux probabilités dans l'application 

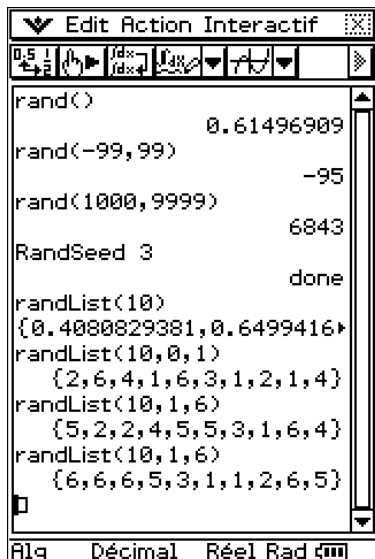
Commençons par les fonctions qui permettent de générer des nombres aléatoires.

– L'expression `rand()` (ici sans argument) renvoie un nombre aléatoire compris entre 0 et 1 (utiliser le mode décimal ou l'icône  pour que le résultat soit un réel en virgule flottante). L'expression `rand(a,b)` où a et b sont des entiers, renvoie un entier aléatoire compris (au sens large) entre a et b .

– L'expression `randList(n)`, où n est un entier strictement positif, renvoie une liste de n réels aléatoires de l'intervalle $]0, 1[$. L'expression `randList(n,a,b)` renvoie une liste de n entiers aléatoires compris entre a et b .

Par exemple `randList(n,0,1)` et `randList(n,1,6)` permettent de modéliser n lancers d'une pièce (0 pour face, 1 pour pile) ou d'un dé à six faces (fig19).

– La procédure `RandSeed` qui prend un argument entier compris entre 0 et 9 peut servir à recalculer le générateur de nombres aléatoires, comme on le voit (fig20).

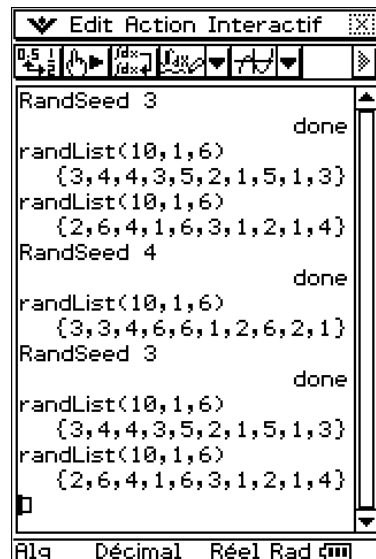


```

Edit Action Interactif
rand()
0.61496909
rand(-99,99)
-95
rand(1000,9999)
6843
RandSeed 3
done
randList(10)
{0.4080829381,0.6499416}
randList(10,0,1)
{2,6,4,1,6,3,1,2,1,4}
randList(10,1,6)
{5,2,2,4,5,5,3,1,6,4}
randList(10,1,6)
{6,6,6,5,3,1,1,2,6,5}
Alg Décimal Réel Rad

```

fig19 : rand et randList

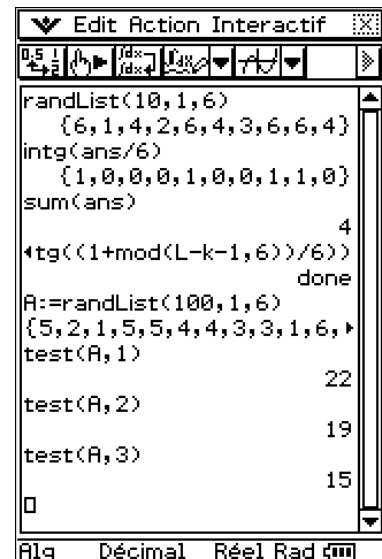


```

Edit Action Interactif
RandSeed 3
done
randList(10,1,6)
{3,4,4,3,5,2,1,5,1,3}
randList(10,1,6)
{2,6,4,1,6,3,1,2,1,4}
RandSeed 4
done
randList(10,1,6)
{3,3,4,6,6,1,2,6,2,1}
RandSeed 3
done
randList(10,1,6)
{3,4,4,3,5,2,1,5,1,3}
randList(10,1,6)
{2,6,4,1,6,3,1,2,1,4}
Alg Décimal Réel Rad

```

fig20 : RandSeed



```

Edit Action Interactif
randList(10,1,6)
{6,1,4,2,6,4,3,6,6,4}
intg(ans/6)
{1,0,0,0,1,0,0,1,1,0}
sum(ans)
4
↑tg((1+mod(L-k-1,6))/6)
done
A:=randList(100,1,6)
{5,2,1,5,5,4,4,3,3,1,6,}
test(A,1)
22
test(A,2)
19
test(A,3)
15
Alg Décimal Réel Rad

```



fig21 : une fonction test

La fonction `randList` permet de simuler n lancers successifs d'un dé à six faces, mais il peut être intéressant de savoir combien de fois chacun des chiffres est apparu dans cette liste.

Pour cela, il faut créer sa propre fonction.

Si L est une liste d'entiers compris entre 1 et 6, alors `intg(L/6)` renvoie une liste où tous les entiers initiaux sont remplacés par 0 (sauf 6, qui est remplacé par 1). La somme de cette liste renvoie donc le nombre de 6 qui figuraient dans la liste initiale. Avec un peu de réflexion, on voit que la fonction nommée `test` et définie par `test(L,k)=sum(intg((1+mod(L-k-1,6))/6))` renvoie le nombre de fois où k est présent dans la liste L (elle-même supposée ne contenir que des entiers de 1 à 6).

On voit (fig21) la création d'une liste aléatoire de 100 entiers compris entre 1 et 6 (ce qui représente 100 lancers d'un dé honnête à 6 faces) et on voit que ce dé a renvoyé 22 fois le résultat 1, 19 fois le résultat 2, et 15 fois le résultat 3.

Dans , on dispose d'un certain nombre de fonctions prédéfinies, notamment pour calculer la loi, la densité ou la fonction de répartition de variables aléatoires réelles suivant des lois discrètes ou continues classiques. Nous n'évoquerons ici que quelques exemples (et nous rappelons que la fenêtre **Probabilités**, accessible à partir de , permet de faire ces calculs de façon plus interactive).

- L'instruction `binomialPD k,n,p` (resp. `binomialCD k,n,p`) (P pour probabilité, C pour cumul et D pour distribution), calcule la probabilité pour qu'une variable aléatoire suivant la loi $\mathcal{B}(n,p)$ prenne la valeur k (resp. une valeur inférieure ou égale à k).

Le résultat est placé dans la variable-système `prob`.


On voit par exemple (fig22) que la probabilité d'obtenir exactement 3 succès (resp. au plus 3 succès) après 10 tentatives indépendantes (la probabilité de réussir à une tentative donnée étant 0,4) vaut approximativement 0,215 (resp. 0,382).

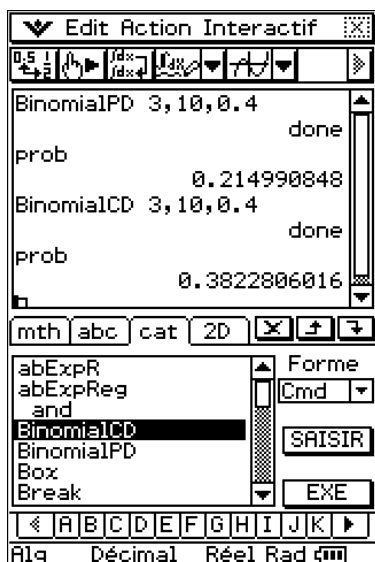
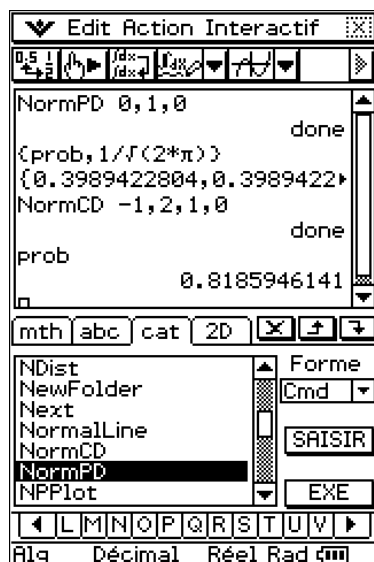
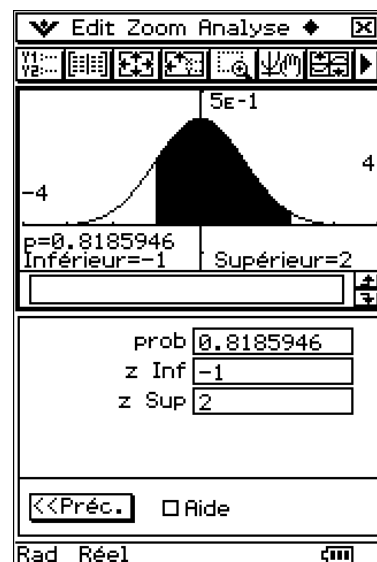
- L'instruction `NormPD x,σ,μ` calcule la densité au point x d'une v.a.r suivant la loi normale $\mathcal{N}(\mu,\sigma)$ (où μ est l'espérance et σ l'écart-type).

L'instruction `NormCD a,b,σ,μ` calcule la probabilité que cette même v.a.r prenne ses valeurs dans l'intervalle $[a,b]$.

Le résultat est placé dans la variable-système `prob`.

On voit par exemple (fig23) que pour la loi $\mathcal{N}(0,1)$ (donc la loi normale centrée réduite : ici $m = 0$ et $\sigma = 1$), la densité au point $1/2$ vaut $\approx 0,399$ c'est-dire $1/\sqrt{(2\pi)}$ comme on le vérifie aisément. On voit également que la probabilité pour qu'une variable suivant cette loi prenne ses valeurs dans $[-1,2]$ vaut $\approx 0,819$.

De tels calculs sont plus faciles à conduire dans l'environnement  (cf fig24), mais on peut avoir besoin des fonctions `NormCD`, `NormPD`, etc. dans un programme, par exemple.

fig22 : `binomialPD` (CD)fig23 : `NormPD` et `NormCD`fig24 : Loi normale $\mathcal{N}(0,1)$

Le Classpad connaît bien sûr beaucoup d'autres lois classiques !

Il est également possible d'utiliser les calculs sur les listes, ou bien l'environnement « tableur », pour faire des calculs de probabilité (nous utiliserons ces possibilités à l'occasion d'autres exercices proposés à l'Oral 2 du Capes de mathématiques). Nous en resterons donc là pour cette présentation des probabilités sur le Classpad.