Atelier Stratégies d'enseignement

Niveau : Spécialité mathématiques en terminale

Complément sur les problèmes « Combinatoire et dénombrement »

La planche de Galton

Dans la planche de Galton, plusieurs billes tombent au travers d'une pyramide de clous sur une planche inclinée. En bas se trouvent des boîtes dans lesquelles tombent les billes. Chaque fois qu'une bille tape un clou, elle poursuit sa chute de manière équiprobable à gauche ou à droite. La bille finit sa trajectoire en tombant dans une des boîtes du bas.
Dans l'exemple ci-contre, il y a 9 cases en bas.

Combien de chemins différents une bille peut-elle parcourir ?

Quelle est la probabilité qu'une bille tombe tout à gauche ? La probabilité qu'une bille tombe dans la case centrale est-elle la même ?

Histoire : *Sir Francis Galton (1822-1911) est un anthropologue, explorateur, géographe, inventeur, météorologue, écrivain, proto-généticien, psychométricien et statisticien britannique. Il est considéré comme le fondateur de l'eugénisme. Il est également connu pour avoir mis en place de façon systématique la méthode d'identification des individus au moyen de leurs empreintes digitales.*

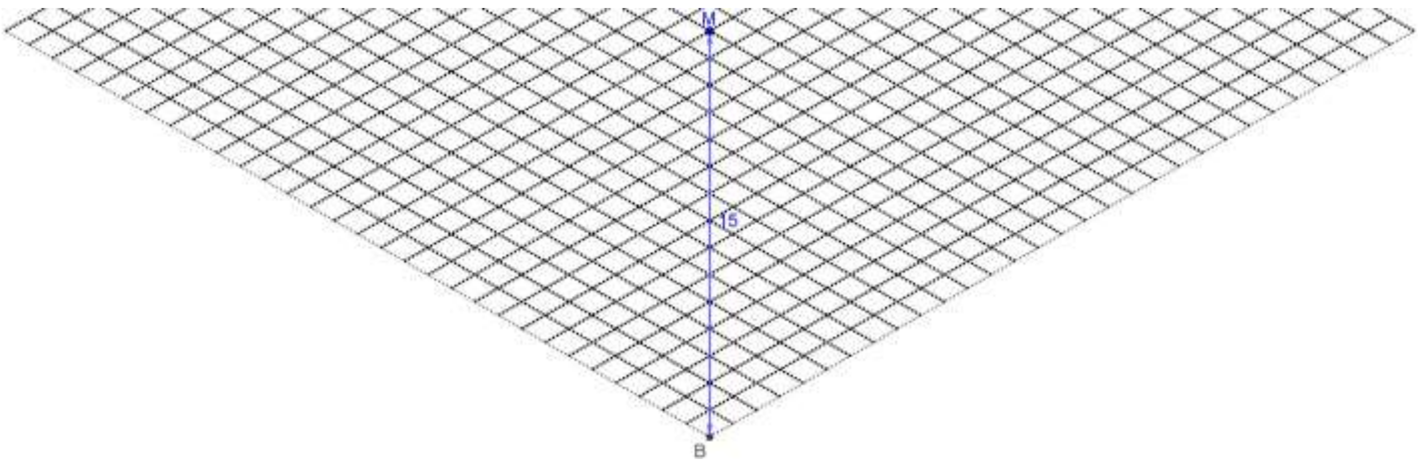
Éléments de correction

Le nombre de chemins est : $2^8 = 256$

La probabilité d'aller à gauche est $\frac{1}{256}$: celle d'aller au centre est $\frac{\binom{8}{4}}{2^8} = \frac{70}{256}$

Exercice 2 :

Un enfant situé au point B et veut rentrer chez lui. Sa maison (point Maison) n'est qu'à 15 pas de B s'il y va en ligne droite. Mais il ne réussit pas à marcher droit. Chaque pas qu'il fait vers l' « avant » (en direction de sa maison) est de façon équiprobable soit vers la droite soit vers la gauche (en suivant les pointillés).



1) L'enfant effectue 30 pas. Déterminer la probabilité qu'il rentre chez lui.
2) Contrôler votre résultat grâce à un programme Python (on pourra modifier l'un des programmes précédents).

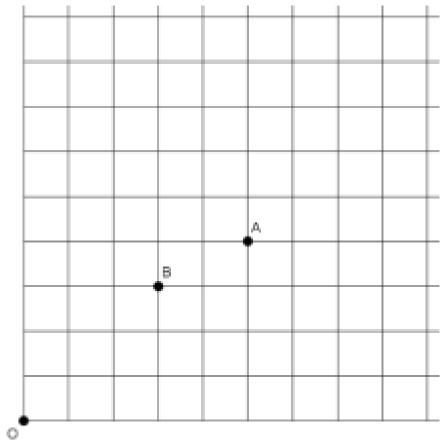
[Aide personnalisé] Que se passe-t-il si on fait plus de pas à gauche que de pas à droite ?

Eléments de correction :

Autant de pas à gauche qu'à droite donc $\binom{30}{15}$ / 2^30 ≈ 0,1444

Exercice 3 : Déplacement d'un robot sur une grille

➔ Un robot est positionné au point O. Il peut se déplacer uniquement sur le quadrillage vers la droite ou vers le haut. Une étape de déplacement correspond à un carreau.



Question 1

En 9 étapes :

1. a. Combien de chemins permettent d'aboutir au point A ?
   b. Quelle est la probabilité que le robot arrive en A ?
2. Combien de chemins passent par le point B ?

[Pour les rapides] Combien de chemins passe par B et aboutissent en A.

Question 2

La liste liste_des_chemins contient tous les chemins en *n* étapes.

Par exemple, en 5 étapes, un élément de la liste peut-être 'DDHDH' qui signifie déplacement à droite aux deux premières étapes, puis en haut, puis à droite et enfin en haut.

Compléter le script qui permet de compter le nombre de chemins passant par le point $M(a,b)$

Programme Python

```
def nombre_de_chemins_passant_par(liste_des_chemins,a,b):
    total = 0
    for chemin in liste:
        x,y=...,...
        for i in chemin:
            if i=='D':
                ...
            else:
                ...
            if (.........) and (.........):
                ...
    return total
```

Bilan donnant lieu à une trace écrite :

Projection (écrite ou vidéo) d'une production d'élève ou co-écriture d'une correction	**Les différentes étapes de la question 2 à travers les 6 compétences : on pourra utiliser un jeu de couleurs**	**Auto-évaluation**
Éléments de correction 1.a. Pour aboutir en A il faut et il suffit de se déplacer 5 fois sur la droite et 4 fois vers le haut.	**Chercher** : • Démarrer la recherche • Valider ou invalider la représentation par les k parmi n ou des puissances, s'ils n'ont pas été trouvés en autonomie	**Aidé :** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Représenter : Reconnaitre un p-uplets	**☺ ou ☹**

Dans les 9 étapes, il y a 5 déplacements à droite soit $\binom{9}{5} = 126$ chemins possibles. 1.b. Il y a $2^9 = 512$ chemins en 9 étapes. Tous les chemins sont équiprobables, donc $p = \frac{126}{512} = \frac{63}{256}$ 2. Il y a 6 étapes pour aller en B, en 6 étapes il y a $\binom{6}{3} = 20$ chemins passant par B en 6 étapes. Les trois dernières étapes n'ont pas de contraintes d'où $20 \times 2^3 = 160$ chemins passant par B en 9 étapes Bonus : $\binom{6}{3} \times \binom{3}{2} = 60$ chemins	*Cas d'une situation avec algorithme* Passer de la situation aux listes	
	Modéliser : Reconnaitre un problème de dénombrement Traduire la situation de dénombrement par un k parmi n	☺ ou ☹
	Calculer : Calculer $\binom{n}{k}$	☺ ou ☹
	Communiquer : Vocabulaire adapté : « k parmi n» et écritures justes	☺ ou ☹

Correction de la partie programmation

Programme calcul_de_chemins.py

```
def ajouter_une_etape(liste_chemins):
    l = len(liste_chemins)
    if l==0:
        liste_chemins=['D','H']
    else:
        for i in range(l):
            liste_chemins.append(liste_chemins[i]+'H')
        for i in range(l):
            liste_chemins[i]=liste_chemins[i]+'D'
    return liste_chemins

def creation_chemins(nombre_d_etapes):
    liste_chemins = []
    for i in range(nombre_d_etapes):
        liste_chemins = ajouter_une_etape(liste_chemins)
    return liste_chemins
```

Programme principal (on peut fournir directement le programme précédent aux élèves sans qu'ils aient à l'ouvrir)

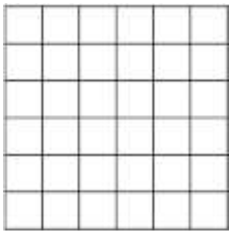
```
from calcul_de_chemin import *

def nombre_de_chemins_passant_par(liste, abscisse, ordonnee):
    total = 0
    for chemin in liste:
        x,y=0,0
        for i in chemin:
            if i=='D':
                x=x+1
            else:
                y=y+1
            if (x==abscisse) and (y==ordonnee):
                total = total + 1
    return total
```

Bilan de la partie algorithmique donnant lieu à une trace écrite :

Projection et exécution du code	**Les capacités algorithmiques**	**Auto-évaluation**
	Lire du code	**En paniquant …** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Appréhender une nouvelle instruction,	☺ **ou** ☹
	Comprendre le rôle des variables	☺ **ou** ☹
	Boucle	☺ **ou** ☹
	Bloc conditionnel	☺ **ou** ☹
	Utilisation d'un bloc fonctionnel	☺ **ou** ☹

<u>Exercice 4</u> : Pixels et images RVB

➔ **Quadrillage 6×6**



On affiche des images à partir du quadrillage ci-contre de taille 6×6 dont chaque case colorée en rouge, vert ou bleu représente un pixel.

1. Dire si l'affirmation suivante est vraie ou fausse : *« on peut afficher plus de cent millions de milliards d'images différentes »*.
2. Combien d'images différentes comprenant exactement 10 pixels rouges peut-on afficher ?
3. Combien d'images différentes comprenant exactement 10 pixels verts et 10 pixels bleus peut-on afficher ?
4. Un pixel est défectueux et on ne voit pas sa couleur.
   Combien d'images différentes comprenant exactement 10 pixels rouges peut-on alors afficher ?
   Expliquer votre démarche à votre binôme qui validera ou invalidera la compréhension de chaque étape (éventuellement par une vidéo réalisée hors temps scolaire).
5. Lire le code Python ci-dessous et déterminer le rôle de la fonction Python « mystere ».

Programme

```
from random import*

def image():
    L=[]
    for i in range(36) :
        L=L+[choice("RVB")]
        Return L

def nb_rouge() :
    nb=0
    L=image()
    for i in range(36) :
        if L[i]=='R' :
            nb=nb+1
    return nb
```

```
def mystère() :
    S=0
    for i in range(10000) :
       if nb_rouge()==10 :
          S=S+1
    return S/10000
```

Info Python
L'instruction `choice()` permet de tirer une lettre au hasard.

6. Tester cette fonction « mystere » et indiquer si les résultats affichés sont cohérents avec votre réponse à la question 2.

➔ **Différenciation :**

- Question 2 remplacée par : Combien d'images différentes comprenant exactement 35 pixels rouges peut-on afficher ?
- Quadrillage 4×4
- Avec seulement 2 couleurs : noir et blanc
- Soient n, k et l des entiers tels que $0 \leq k \leq n$, $0 \leq l \leq n$ et $k + l \leq n$.
  Quadrillage $n \times n$ et déterminer le nombre d'images différentes comprenant exactement k pixels verts et l pixels bleus que l'on peut afficher
- Faire programmer les fonctions à l'aide d'algorithmes plus ou moins complets et remplacer la question 5 par : Compléter les fonctions Python suivantes pour que la fonction freq_rouges() renvoie la renvoie la fréquence d'images ayant exactement 10 pixels rouges parmi $10\ 000$ simulations d'images de taille 6×6.

Bilan donnant lieu à une trace écrite :

Projection (écrite ou vidéo) d'une production d'élève ou co-écriture d'une correction	**Les différentes étapes de la question 2 à travers les 6 compétences : on pourra utiliser un jeu de couleurs**	**Auto-évaluation**
Éléments de correction 1. $3^{36} \approx 1{,}5 \times 10^{17}$ 2. $\binom{36}{10} \times 2^{26}$ 36 uplet : _ _ _ _ _ _ _ _ _ _ _ _ _ 10 rouges parmi 36 cases les autres étant vertes ou bleues 3. $\binom{36}{10} \times \binom{26}{10}$ $\binom{35}{10} \times 2^{25}$	**Chercher** : • Démarrer la recherche • Valider ou invalider la représentation par les k parmi n ou des puissances, s'ils n'ont pas été trouvés en autonomie	**Aidé :** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Représenter : Reconnaitre un p-uplets *Cas d'une situation avec algorithme* Passer de la situation aux listes	☺ **ou** ☹
	Modéliser : Reconnaitre un problème de dénombrement Traduire la situation de dénombrement par un k parmi n	☺ **ou** ☹
	Calculer : Calculer $\binom{n}{k}$	☺ **ou** ☹
	Communiquer : Vocabulaire adapté : « k parmi n» et écritures justes	☺ **ou** ☹

Bilan de la partie algorithmique donnant lieu à une trace écrite :

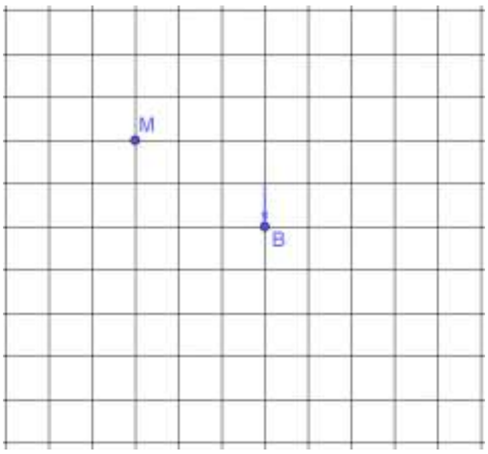
Projection et exécution du code	**Les capacités algorithmiques**	**Auto-évaluation**
Éléments de correction 4. La fonction « mystere » renvoie la fréquence d'images ayant exactement 10 pixels rouges parmi 10000 simulations d'images de taille 6×6. 5. On obtient des fréquences proches de $\binom{36}{10} \times \frac{2^{26}}{3^{36}} \approx 0{,}11365$ qui est la proportion d'images de taille 6×6 comptant exactement 10 pixels rouges.	Lire du code	**En paniquant …** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Appréhender une nouvelle instruction,	☺ **ou** ☹
	Comprendre le rôle des variables	☺ **ou** ☹
	Boucle	☺ **ou** ☹
	Bloc conditionnel	☺ **ou** ☹
	Utilisation d'un bloc fonctionnel	☺ **ou** ☹

<u>Exercice 5</u> : Manhattan

À Manhattan, les routes forment un quadrillage à angle droit. Les avenues sont les lignes verticales et les rues les lignes horizontales du plan.
Manhattan peut ainsi être modélisé par un quadrillage régulier, où chaque point est l'intersection d'une avenue et d'une rue et est relié à chacun de ses 4 voisins par un segment de longueur 1.
Un adolescent, situé au point B (son lycée) et orienté vers le bas, veut rentrer chez lui (point M, maison). Il parcourt au total un chemin de longueur 5, en tournant, à chaque intersection, de manière équiprobable à gauche ou à droite.

1) À quoi sert l'algorithme ci-dessous ?
2) Quelle est la probabilité que l'adolescent repasse sur le point B ?
Aide : Combien de chemins différents peut-il prendre ?
3) Déterminer un chemin permettant à l'adolescent de rentrer chez lui. *On admet momentanément l'unicité de ce chemin.*
4) Quelle est la probabilité que l'adolescent rentre chez lui ?
5) Sachant qu'il a tourné au moins deux fois à gauche, quelle est la probabilité que l'adolescent soit rentré chez lui ?
Aide : Supposer dans un premier temps qu'il a tourné exactement deux fois à gauche.
Aide ou remarque: Faire le lien avec la preuve de $\sum_{k=0}^{n}\binom{n}{k} = 2^n$.

6) *Construire un programme Python pour retrouver les résultats précédents et prouver l'unicité admise à la question 3).



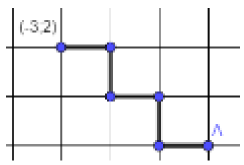
Programmer avec Turtle

```
from turtle import*
from random import*

def Chemins(n):
    for i in range(n):
        up()
        goto(0,0)
        down()
        speed(0)
        for i in range(5):
            if choice("DG")=="D":
                right(90)
            else:
                left(90)
```

Info Python

L'instruction `choice()` permet de tirer une lettre au hasard.

Bilan donnant lieu à une trace écrite :

Projection (écrite ou vidéo) d'une production d'élève ou co-écriture d'une correction	**Les différentes étapes à travers les 6 compétences : on pourra utiliser un jeu de couleurs**	**Auto-évaluation**
Éléments de correction 1. Simule un « chemin aléatoire » de longueur 5 2. On démontre qu'il y a exactement 4 chemins de longueur 4 permettant de revenir en B. Multiplié par les 2 possibilités pour le dernier choix : 8 chemins possibles 3. Donc la probabilité cherchée vaut $\frac{8}{2^5}$ 4. Chemin :  5. $\frac{1}{2^5}$ 6. $\frac{1}{\binom{5}{2}+\binom{5}{3}+\binom{5}{4}+\binom{5}{5}}$	**Chercher** : • Démarrer la recherche • Valider ou invalider la représentation par les k parmi n ou des puissances, s'ils n'ont pas été trouvés en autonomie	**Aidé :** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Représenter : Reconnaitre un p-uplets *Cas d'une situation avec algorithme* Passer de la situation aux listes	☺ **ou** ☹
	Modéliser : Reconnaitre un problème de dénombrement Traduire la situation de dénombrement par un k parmi n	☺ **ou** ☹
	Calculer : Calculer $\binom{n}{k}$	☺ **ou** ☹
	Communiquer : Vocabulaire adapté : « k parmi n» et écritures justes	☺ **ou** ☹

Bilan de la partie algorithmique donnant lieu à une trace écrite :

<table>
<tr><th>Projection et exécution du code</th><th>Les capacités algorithmiques</th><th>Auto-évaluation</th></tr>
<tr><td rowspan="6"><pre>
from turtle import*
from random import*

def Chemins(n):
    for i in range(n):
        up()
        goto(0,0)
        down()
        speed(0)
        for i in range(5):
            if choice("DG")=="D":
                right(90)
            else:
                left(90)
            forward(100)
mainloop()
</pre></td><td>Lire du code</td><td>En paniquant …
☐Pas du tout
☐Un peu
☐Beaucoup
☐Enormément</td></tr>
<tr><td>Appréhender une nouvelle instruction,</td><td>☺ ou ☹</td></tr>
<tr><td>Comprendre le rôle des variables</td><td>☺ ou ☹</td></tr>
<tr><td>Boucle</td><td>☺ ou ☹</td></tr>
<tr><td>Bloc conditionnel</td><td>☺ ou ☹</td></tr>
<tr><td>Utilisation d'un bloc fonctionnel</td><td>☺ ou ☹</td></tr>
</table>

<u>Exercice 6</u> : L'affaire « États-Unis contre Tucker »

➔ Dans l'affaire « États-Unis contre Tucker », en 1983, 35 jurés furent présélectionnés pour constituer le jury, parmi lesquels 4 Noirs.
Chaque camp peut écarter 7 jurés et, parmi les 7 choisis par le gouvernement des États-Unis figurèrent les 4 Noirs. Accusé de discrimination, le procureur se défendit en affirmant qu'il « désirait des jurés ayant fait des études supérieures et que la couleur de la peau n'était en aucune manière intervenue pour motiver les récusations ». En considérant les 35 présélectionnés, il apparait que 18 n'ont pas fait d'études supérieures dont les 4 Noirs.

(D'après H. Zeilel et D. Kaye *Prove it with Figures*)

<u>Modélisation</u>

On considère une urne contenant 18 boules : les boules 1, 2, 3 et 4 sont noires et les boules 5 à 18 sont blanches.
On prélève au hasard et sans remise, 7 boules.
Evaluation de la probabilité de l'évènement E : « Parmi les 7 boules prélevées figurent les boules 1 à 4 »

1. On souhaite écrire un programme effectuant une boucle de 100 expériences et affichant la fréquence de réalisation de l'évènement E parmi ces 100 expériences. Lequel des deux programmes ci-dessous convient-il ? justifier.

Info Python
L'instruction
`random.sample(range(1,19),7)`
permet de réaliser un tirage sans remise de 7 éléments parmi les entiers de 1 à 18(exclu).

<table>
<tr><th>Programme 1</th><th>Programme 2</th></tr>
<tr><td><pre>
import random

def prob() :
    e=0
    for i in range(100):
        T=random.sample(range(1,19),7)
        if 1 in T and 2 in T and 3 in T and 4 in T:
            e=e+1
    return e
</pre></td><td><pre>
import random

def prob() :
    e=0
    for i in range(100):
        T=random.sample(range(1,19),7)
        for k in[1,2,3,4]:
            if k in T:
                e=e+1
    return e
</pre></td></tr>
</table>

2. Implémenter ce programme et estimer la probabilité de E en l'exécutant plusieurs fois.
3. Que faut-il modifier dans ce programme pour que la fréquence soit plus stable d'une simulation à l'autre ?

<u>Etude théorique</u>

1. Quel est le nombre de listes de 7 boules parmi les 18 que l'on peut former ?
2. Quel est le nombre de listes de 7 boules où figurent les boules 1 à 4 ?
3. En déduire la fréquence de l'évènement E.

<u>Conclusion</u>

Pourquoi est-on en droit de demander au procureur une explication plus convaincante que celle fournie ?

Bilan donnant lieu à une trace écrite :

<table>
<tr><th>Projection et exécution du code</th><th>Les capacités algorithmiques</th><th>Auto-évaluation</th></tr>
<tr><td rowspan="6">Éléments de correction
Programmation
1. Le programme qui convient est celui de gauche car on y incrémente e lorsque les quatre boules 1, 2, 3 et 4 sont tirées (utilisation de « and »). Dans le programme de droite, on incrémente e dès que l'une des boules 1, 2, 3 ou 4 est tirées.
2. On peut estimer la probabilité de E à 0,01 mais la fluctuation est importante.
3. On augmente le nombre d'expériences simulées en remplaçant par exemple 100 par 10 000 dans la ligne 4. On peut estimer la probabilité de E à 0,011.</td><td>Lire du code</td><td>En paniquant …
☐Pas du tout
☐Un peu
☐Beaucoup
☐Enormément</td></tr>
<tr><td>Appréhender une nouvelle instruction,</td><td>☺ ou ☹</td></tr>
<tr><td>Comprendre le rôle des variables</td><td>☺ ou ☹</td></tr>
<tr><td>Boucle</td><td>☺ ou ☹</td></tr>
<tr><td>Bloc conditionnel</td><td>☺ ou ☹</td></tr>
<tr><td>Utilisation d'un bloc fonctionnel</td><td>☺ ou ☹</td></tr>
</table>

Bilan de la partie algorithmique donnant lieu à une trace écrite :

Projection (écrite ou vidéo) d'une production d'élève ou co-écriture d'une correction	Les différentes étapes de la question 2 à travers les 6 compétences : on pourra utiliser un jeu de couleurs	Auto-évaluation
Etude théorique 1. $\binom{18}{7} = 31824$ 2. $\binom{14}{3} = 364$ 3. $\frac{364}{31824} \approx 0{,}0114$ **Conclusion** La présence des 4 jurés noirs parmi les 7 personnes écartées par le procureur, choisies parmi les 18 n'ayant pas fait d'études supérieures, peut être le fait du hasard ; mais la probabilité de cet événement étant estimée à 1/100, cela est très improbable ! Toute laisse penser qu'il aura fallu « aider » le hasard pour parvenir à un tel résultat.	**Chercher** : • Démarrer la recherche • Valider ou invalider la représentation par les k parmi n ou des puissances, s'ils n'ont pas été trouvés en autonomie	**Aidé :** ☐**Pas du tout** ☐**Un peu** ☐**Beaucoup** ☐**Enormément**
	Représenter : Reconnaitre un p-uplets *Cas d'une situation avec algorithme* Passer de la situation aux listes	☺ **ou** ☹
	Modéliser : Reconnaitre un problème de dénombrement Traduire la situation de dénombrement par un k parmi n	☺ **ou** ☹
	Calculer : Calculer $\binom{n}{k}$	☺ **ou** ☹
	Communiquer : Vocabulaire adapté : « k parmi n» et écritures justes	☺ **ou** ☹