

Dénombrement**Fiche Défi**

Nom : _____

Date : _____

Exercice D1 : Symétrie et relation de Pascal

Soit $n \geq 1$ un entier. On admet, pour $0 \leq k \leq n$, que $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.

- Par le calcul.** Pour $0 \leq k \leq n-1$, montrer que $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.
- Par dénombrement.** Pour $0 \leq k \leq n-1$, soit F un ensemble à $n+1$ éléments et $a \in F$. On note $\mathcal{P}_{k+1}$ l'ensemble des parties de F à $k+1$ éléments. Séparer les parties selon qu'elles contiennent a ou non, et retrouver la relation de Pascal.
- Symétrie.** Soit E un ensemble à n éléments et $0 \leq k \leq n$. À toute partie A de E à k éléments, on associe son complémentaire $\bar{A}$. Montrer que cette application est une bijection de l'ensemble des parties à k éléments sur l'ensemble des parties à $n-k$ éléments, et en déduire $\binom{n}{k} = \binom{n}{n-k}$.
- Python.** Compléter la fonction suivante, qui construit la ligne n du triangle de Pascal à partir de la ligne précédente, puis donner l'affichage.

```
def ligne(n):
    L = [1]
    for i in range(n):
        L = [1] + [... for j in range(len(L) - 1)] + [1]
    return L
print(ligne(6))
```

Exercice D2 : Triangle de Pascal et suite de Fibonacci

Pour tout entier $n \geq 0$, on pose $u_n = \sum_{k=0}^n \binom{n-k}{k}$, avec la convention $\binom{a}{b} = 0$ lorsque $b > a$. Par exemple, $u_4 = \binom{4}{0} + \binom{3}{1} + \binom{2}{2} + \binom{1}{3} + \binom{0}{4}$, où les deux derniers termes sont nuls. On admet que la relation de Pascal $\binom{m+1}{k} = \binom{m}{k} + \binom{m}{k-1}$ reste vraie avec cette convention, pour tous entiers $m \geq 0$ et $k \geq 1$.

- Calculer $u_0, u_1, \dots, u_6$.
- Soit $n \geq 0$. Montrer que $u_{n+2} = u_{n+1} + u_n$. (Écrire u_{n+2} en isolant le terme d'indice $k=0$; vérifier que le terme d'indice $k=n+2$ est nul; appliquer la relation de Pascal aux termes d'indices $1 \leq k \leq n+1$; reconnaître u_{n+1} et u_n .)
- On définit la suite (F_n) par $F_0 = 0$, $F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$. Démontrer par récurrence que, pour tout entier $n \geq 0$, $u_n = F_{n+1}$.
- Un escalier comporte n marches. On le monte en franchissant à chaque pas une ou deux marches. Soit k le nombre de pas de deux marches utilisés. Combien y a-t-il de pas au total? Justifier que le nombre de façons de monter l'escalier avec exactement k pas de deux marches est $\binom{n-k}{k}$, puis que le nombre total de façons de monter l'escalier est u_n . Donner ce nombre pour $n=5$.
- Voici un script.

```
from math import comb
def u(n):
    return sum(comb(n-k, k) for k in range(n+1))
print([u(n) for n in range(7)])
print(u(20))
```

Donner les deux affichages (on admet que `comb(a, b)` renvoie 0 si $b > a$).

Exercice D3 : Les parties d'un ensemble

Soit $E = \{x_1; x_2; \dots; x_n\}$ un ensemble à $n \geq 1$ éléments.

- À une partie A de E , on associe le n -uplet $(\varepsilon_1, \dots, \varepsilon_n)$ de $\{0; 1\}^n$ défini par $\varepsilon_i = 1$ si $x_i \in A$ et $\varepsilon_i = 0$ sinon. Montrer que cette application est une bijection et en déduire que E possède 2^n parties.
- On admet qu'il y a $\frac{n!}{(n-k)!}$ k -uplets d'éléments deux à deux distincts de E (pour $0 \leq k \leq n$). En comptant de deux façons ces k -uplets, montrer que le nombre de parties à k éléments est $\frac{n!}{k!(n-k)!}$.
- En déduire $\sum_{k=0}^n \binom{n}{k} = 2^n$.
- On fixe $a \in E$. Pour une partie A de E , on pose $\varphi(A) = A \setminus \{a\}$ si $a \in A$ et $\varphi(A) = A \cup \{a\}$ sinon. Montrer que $\varphi \circ \varphi$ est l'identité et que φ change la parité du cardinal. En déduire qu'il y a autant de parties de cardinal pair que de parties de cardinal impair, et donner ce nombre.
- Le script ci-dessous énumère les parties de $\{0, 1, 2, 3, 4\}$ à l'aide des mots binaires. Qu'affiche-t-il ? Quel lien avec les coefficients binomiaux ?

```
n = 5
effectifs = [0] * (n + 1)
for m in range(2 ** n):
    mot = format(m, '05b')
    effectifs[mot.count('1')] += 1
print(effectifs)
```

Exercice D4 : Un double comptage

Une classe comporte n filles et n garçons ($n \geq 1$). On forme un groupe de n élèves.

- Combien de groupes peut-on former ?
- Pour $0 \leq k \leq n$, montrer que le nombre de groupes comportant exactement k filles est $\binom{n}{k}^2$ (on utilisera la symétrie des coefficients binomiaux).
- En déduire $\sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}$ et vérifier cette égalité pour $n = 5$.
- Pour aller plus loin, hors programme.* Plus généralement, une classe comporte m filles et p garçons et on forme un groupe de r élèves, avec $r \leq m + p$. Montrer que $\binom{m+p}{r} = \sum_{k=0}^r \binom{m}{k} \binom{p}{r-k}$, avec la convention $\binom{a}{b} = 0$ si $b > a$. Vérifier pour $m = 7$, $p = 5$, $r = 4$.
- Pour aller plus loin, hors programme.* Soit E un ensemble à n éléments et $1 \leq k \leq n$. En comptant de deux façons les couples (A, c) où A est une partie de E à k éléments et $c \in A$, démontrer que $k \binom{n}{k} = n \binom{n-1}{k-1}$. En déduire $\sum_{k=0}^n k \binom{n}{k} = n 2^{n-1}$ et vérifier pour $n = 5$.

Exercice D5 : Réunion, cas disjoints et applications surjectives

- Soient A et B deux ensembles finis. En écrivant $A \cup B$ comme réunion de deux ensembles disjoints, puis B comme réunion de deux ensembles disjoints, démontrer que $\text{Card}(A \cup B) = \text{Card}(A) + \text{Card}(B) - \text{Card}(A \cap B)$.
- Combien d'entiers de 1 à 100 ne sont divisibles ni par 2, ni par 3 ?
- Parmi ces entiers, combien sont multiples de 5 ? (On pourra appliquer la question 1 à l'ensemble des multiples de 5 divisibles par 2 et à celui des multiples de 5 divisibles par 3.) En déduire combien d'entiers de 1 à 100 ne sont divisibles ni par 2, ni par 3, ni par 5.
- Une application de $\{1; 2; 3; 4; 5\}$ dans $\{a; b; c\}$ est dite surjective si chacune des lettres a, b, c est l'image d'au moins un élément. Pour une application f , on note $(n_a; n_b; n_c)$ les nombres d'éléments

dont l'image est respectivement a, b, c .

a) Pour un triplet $(n_a; n_b; n_c)$ donné, montrer que le nombre d'applications correspondantes est

$$\binom{5}{n_a} \binom{5-n_a}{n_b}.$$

b) Quels triplets $(n_a; n_b; n_c)$ d'entiers supérieurs ou égaux à 1 de somme 5 existe-t-il ? En déduire le nombre d'applications surjectives.

5. Écrire un script Python qui vérifie ce dernier résultat en énumérant les applications.

Exercice D6 : Répartitions de jetons

On répartit 10 jetons identiques entre trois joueurs A, B, C . Une répartition est un triplet (x, y, z) d'entiers naturels tel que $x + y + z = 10$ (x, y, z désignent les nombres de jetons de A, B, C).

1. On représente une répartition par un mot de 12 symboles formé de 10 symboles $\circ$ et de 2 barres $|$, les barres séparant les jetons des trois joueurs. Par exemple, $(3, 0, 7)$ correspond à $\circ\circ\circ||\circ\circ\circ\circ\circ\circ$. Quelle est la répartition associée au mot $\circ| \circ\circ\circ\circ\circ| \circ\circ\circ$? Justifier que cette correspondance est une bijection et en déduire le nombre de répartitions.
2. Combien de répartitions donnent au moins un jeton à chaque joueur ? (Poser $x' = x - 1, y' = y - 1, z' = z - 1$.)
3. En classant les répartitions selon le nombre x de jetons de A , retrouver le nombre total de répartitions de la question 1.
4. Interpréter ce nombre comme un nombre de chemins d'un quadrillage.
5. Compléter le script ci-dessous pour compter les répartitions de la question 1.

```
compte = 0
for x in range(11):
    for y in range(11):
        for z in range(11):
            if ...:
                compte = compte + 1
print(compte)
```

Exercice D7 : Problème : une grille de loto

Une loterie fonctionne ainsi. Un joueur coche 5 numéros distincts parmi les entiers de 1 à 30. Le tirage officiel consiste à choisir au hasard 5 numéros distincts parmi ces 30 entiers, tous les tirages étant équiprobables. On note X le nombre de numéros communs à la grille du joueur et au tirage.

Partie A.

1. Combien de tirages sont possibles ?
2. Pour $0 \leq k \leq 5$, montrer que le nombre de tirages ayant exactement k numéros communs avec la grille est $\binom{5}{k} \binom{25}{5-k}$. En déduire la loi de X sous forme d'un tableau (valeurs exactes sous la forme de fractions de dénominateur 142 506, puis valeurs approchées à 10^{-4} près).
3. Vérifier que la somme des probabilités vaut 1. Quelle égalité sur les coefficients binomiaux cela traduit-il ?

Partie B. Le joueur verse une mise de 2 euros. Il reçoit 2 euros s'il a 2 numéros communs, 20 euros avec 3, 500 euros avec 4, 10 000 euros avec 5, et rien sinon. On note G le gain brut en euros.

4. Calculer la probabilité de recevoir une somme non nulle.
5. Calculer l'espérance $E(G)$ (valeur exacte, puis valeur approchée à 10^{-2} près). En déduire l'espérance du gain net $G - 2$ et interpréter.
6. Quelle mise rendrait le jeu équitable ?

Partie C. On considère le script suivant.

```

from math import comb
N = comb(30, 5)
gain = {2: 2, 3: 20, 4: 500, 5: 10000}
E = 0
for k in range(6):
    p = comb(5, k) * comb(25, 5 - k) / N
    E = E + gain.get(k, 0) * p
print(E)

```

7. Quelle valeur approchée ce script affiche-t-il ? Quelle modification permet d'obtenir l'espérance du gain net ?

Exercice D8 : Appariements d'un tournoi

Huit joueurs $J_1, \dots, J_8$ sont répartis au hasard en 4 matchs de 2 joueurs. L'ordre des matchs et l'ordre des deux joueurs dans un match ne comptent pas. Un tel découpage en paires est appelé appariement.

1. Montrer qu'il y a $7 \times 5 \times 3 \times 1 = 105$ appariements. (Chercher l'adversaire du joueur de plus petit indice encore sans adversaire.)
2. Retrouver ce nombre en rangeant les huit joueurs dans une file : le premier joue contre le deuxième, le troisième contre le quatrième, etc. Combien de files donnent le même appariement ?
3. Les appariements sont équiprobables. Calculer la probabilité que J_1 et J_2 s'affrontent.
4. Les joueurs $J_1, \dots, J_4$ appartiennent au club X et les autres au club Y . Calculer la probabilité qu'aucun match n'oppose deux joueurs du même club.
5. Le script suivant renvoie la liste des appariements d'une liste de joueurs.

```

def appariements(liste):
    if len(liste) == 0:
        return [[]]
    premier = liste[0]
    resultat = []
    for i in range(1, len(liste)):
        autre = liste[i]
        reste = liste[1:i] + liste[i + 1:]
        for ap in appariements(reste):
            resultat.append([(premier, autre)] + ap)
    return resultat
print(len(appariements(list(range(8)))))

```

Expliquer le lien avec la question 1, donner l'affichage, puis le nombre d'appariements de 10 joueurs.