DM - On compte la plus grande série de succès.

I) Préambule.

Soit n un entier naturel non nul. On considère n lancers identiques et indépendants d'une pièce de monnaie équilibrée.

Soit Y_n la variable aléatoire comptant la plus grande série de faces obtenue sur un échantillon donné. Si l'échantillon ne contient pas de face, alors $Y_n = 0$.

	Échantillons	
$n=4$	$F_1\ P_2\ F_3\ F_4$	$Y_4=2$
$n=4$	$F_1\ F_2\ P_3\ P_4$	$Y_4=2$
$n=3$	$F_1\ F_2\ F_3$	$Y_3=3$
$n=5$	$F_1\ P_2\ F_3\ F_4\ P_5$	$Y_5=2$
$n=7$	$F_1\ F_2\ F_3\ P_4\ F_5\ F_6\ P_7$	$Y_7=3$

L'objectif est d'obtenir des résultats sur la variable aléatoire Y_n.

II) Étude de cas particuliers.

1. Donner $Y_1(\Omega)$ puis établir la loi de probabilité de Y_1. Faire de même pour Y_2.
2. Reconnaître une loi connue pour Y_1 et Y_2 puis établir une conjecture sur la loi de Y_n.
3. Donner $Y_3(\Omega)$ puis la loi de probabilité de Y_3.
4. Que dire de votre conjecture ?
5. Combien y a-t-il d'échantillon de taille n différents ?
6. Déterminer $Y_n(\Omega)$ en donnant un exemple d'échantillon pour obtenir chacune des valeurs.
7. Déterminer $\mathbb{P}(Y_n = 0)$ puis $\mathbb{P}(Y_n = n)$.

III) Modélisation.

L'étude de Y_n est difficile et ne suit pas de loi remarquable. Toutefois cela ne doit pas nous freiner et nous pouvons tout de même obtenir des résultats intéressants.

Pour cela, nous allons modéliser l'ensemble des échantillons de taille n possibles.

Le programme ci-dessous permet de générer l'ensemble des échantillons de taille n. Les échantillons sont rangés dans une liste Python.

Code Python

```
def generation_echantillon(n):
    if p == 0:
        return [""]
    else:
        l = generation_echantillon(n-1)
        nouvelle_liste = []
        for element in l:
            nouvelle_liste.append("P"+element)
            nouvelle_liste.append("F"+element)
        return nouvelle_liste
```

Un échantillon est modélisé à l'aide d'une **chaine de caractères**. Par exemple voici une chaine de caractères modélisant un échantillon de taille 5 :

`"FPPFF"`

On peut parcourir une chaine de caractère à l'aide d'une boucle `for` en utilisant la syntaxe suivante.

`for lettre in echantillon:`

La variable `lettre` prendra la valeur de chacun des caractères de la chaine de caractères.

8. Compléter le programme `compter(echantillon)` permettant de compter la série de faces la plus grande d'un échantillon donné sous la forme d'une chaine de caractères constituée de `"F"` et `"P"`.

Code Python

```
def compter(echantillon):
    compteur = 0
    max = 0
    for lettre in echantillon:
        if lettre == "F":
            compteur += ...
        else:
            compteur = ...
        if ............. :
            max = ...
    return max
```

Il faut finalement un programme qui va permettre de regarder la valeur de Y_n en fonction de tous les échantillons de taille n possibles. On stockera le nombre d'échantillon de taille n permettant de réaliser l'évènement $\{Y_n = k\}$, avec $k \in \{0,\ 1,\ \dots,\ n\}$, dans une liste nommée `occurence`.

On rappelle qu'une liste Python est un **tableau**. Si la liste s'appelle `l`, on peut avoir accès au premier élément en utilisant la syntaxe `l[0]`. Pour avoir accès au deuxième élément on écrit `l[1]` etc...

Voici le script Python permettant d'obtenir l'ensemble des occurences des valeurs de Y_n.

Code Python

```
#Liste de tous les echantillons
epreuve = generation_echantillon(n)

#Liste des occurences des valeurs de Y_n
occurences = [0 for i in range(n+1)]

#Pour chaque echantillon, on regarde la valeur de Y_n
#On ajoute une occurence pour la valeur de Y_n obtenue
for echantillon in epreuve:
    valeur_Y = compter(echantillon)
    occurences[valeur_Y] += 1

print(occurences)
```

Voici les listes d'occurences obtenues pour différentes valeurs de n.

Code Python

```
n = 3 : [1, 2, 1]
n = 4 : [1, 4, 2, 1]
n = 5 : [1, 7, 5, 2, 1]
n = 6 : [1, 12, 11, 5, 2, 1]
n = 7 : [1, 20, 23, 12, 5, 2, 1]
n = 8 : [1, 33, 47, 27, 12, 5, 2, 1]
n = 9 : [1, 54, 94, 59, 28, 12, 5, 2, 1]
n = 10: [1, 88, 185, 127, 63, 28, 12, 5, 2, 1]
```

IV) Conjectures et cas général.

A) Étude du cas $Y_n = 1$.

On note dans cette partie $u_n = \text{Card}\left(\{Y_n = 1\}\right)$ pour tout entier naturel n non nul. u_n représente le nombre d'échantillon dont la série la plus grande de face est égale à 1 face.

1. En vous aidant du tableau des occurences, conjecturer une relation de récurrence double vérifiée par la suite $(u_n)_{n\geq 1}$.
2. Démontrer cette relation de récurrence en utilisant des arguments de dénombrement.

On pose pour tout entier n non nul, $a_n = u_n + 1$.

3. Démontrer que $(a_n)_{n\geq 1}$ vérifie la même relation de récurrence que la suite de Fibonacci. On précisera les deux premiers termes de $(a_n)_{n\geq 1}$.

4. On souhaite démontrer, pour tout entier naturel n non nul, la propriété,

$$(H_n)\ :\ a_n = \frac{\alpha^{n+2} - \beta^{n+2}}{\sqrt{5}}$$

où $\alpha = \dfrac{1+\sqrt{5}}{2}$ et $\beta = \dfrac{1-\sqrt{5}}{2}$ sont les racines du polynôme $X^2 - X - 1$.

Pour démontrer ce résultat, on va utiliser une méthode de démonstration qui est la **récurrence forte**.

Pour cela, on fait une initialisation en montrant que (H_1) et (H_2) sont vraies. Ensuite, dans l'hérédité, au lieu de supposer que la propriété est vraie pour un rang n fixé, on fixe un entier n et on suppose que la propriété est vraie pour tout entier k inférieur ou égal à n.

(a) (Initialisation) Montrer que (H_1) et (H_2) sont vraies.
(b) Justifier que $\alpha^2 = \alpha + 1$ et $\beta^2 = \beta + 1$.
(c) (Hérédité) Soit n un entier fixé. On suppose que pour tout entier k inférieur ou égale à n, la propriété (H_k) est vraie. Montrer que

$$a_{n+1} = \frac{\alpha^{n+3} - \beta^{n+3}}{\sqrt{5}}.$$

5. Exprimer finalement $\mathbb{P}(Y_n = 1)$ en fonction de n.

B) Étude des cas rares : Les grandes séries de succès.

1. Que dire des occurences des valeurs de Y_n pour des k proches de n ? Ces valeurs semblent-elles dépendre de la valeur de n ?
2. Dénombrer le nombre d'échantillon de taille 10 tels que l'évènement $\{Y_{10} = 9\}$ est réalisé.
3. Dénombrer le nombre d'échantillon de taille 10 tels que l'évènement $\{Y_{11} = 10\}$ est réalisé.
4. Y a-t-il une différence dans le raisonnement ? En déduire le nombre d'échantillon de taille n tels que l'évènement $\{Y_n = n-1\}$ est réalisé.
5. En vous aidant d'exemples, dénombrer le nombre d'échantillon de taille n tels que $\{Y_n = n-2\}$ est réalisé.

On remarque que pour des entiers k suffisamment petits tels que $n - k > \frac{n}{2}$, on a

$$\mathrm{Card}(Y_n = n-2) = 2 \times \mathrm{Card}(Y_n = n-1) + 2^0$$

$$\mathrm{Card}(Y_n = n-3) = 2 \times \mathrm{Card}(Y_n = n-2) + 2^1$$

$$\mathrm{Card}(Y_n = n-4) = 2 \times \mathrm{Card}(Y_n = n-3) + 2^2.$$

$$\dots$$

$$(*) \quad \mathrm{Card}(Y_n = n-k) = 2 \times \mathrm{Card}(Y_n = n-k+1) + 2^{k-2}$$

On note pour tout entier k qui convient, $T_k = \mathrm{Card}(Y_n = n-k)$.

6. Démontrer par des arguments de dénombrement que

$$T_k = (k+3)2^{k-2}.$$