Feuille Oraux_3 : Préparation à l'exercice sans préparation de l'agro.

Ex 1 : On travaille sur des matrices M contenant que des 1 et des 0

1) Ecrire une fonction qui une telle matrice M renvoie le nombre de 1 qu'elle contient.
2) Ecrire une fonction qui vérifie sur une telle matrice M qu'aucun de 1 n'a de case voisine contenant un 1. *(On considère comme case voisine uniquement les cases horizontales et verticales, on ne regarde pas en diagonale)*
3) Ecrire une fonction qui vérifie sur une telle matrice M, qu'il y a au moins un 1 sur chaque ligne et au moins 1 sur chaque colonne .

Ex 2 : Dans cet exercice on n'utilise que la fonction `random` du module `random`.

1) Ecrire une fonction qui permet de simuler la réalisation d'une variable suivant la loi uniforme sur $[\![0; n-1]\!]$
2) Ecrire une fonction qui retourne au hasard une permutation de $[\![0; n-1]\!]$.
3) On appelle dérangement une permutation sans point fixe.

   (une permutation $(x_0, ..., x_{n-1})$ telle que $\forall k,\ x_k \neq k$)

   Ecrire une fonction prenant pour argument n et qui renvoie une estimation de la probabilité qu'une permutation de $[\![0; n-1]\!]$ prise au hasard soit un dérangement.

Ex 3 : On travaille sur des matrices M sous forme de liste de listes d'entiers.

1) Ecrire une fonction qui vérifie que toutes listes de M (représentant les lignes) sont de même longueur.
2) Ecrire une fonction qui pour une telle matrice M renvoie `True` si elle est carrée et `False` sinon.
3) Ecrire une fonction qui pour une telle matrice M renvoie sa transposée.

Ex 4 : On travaille ici sur des chaines caractères ne contenant que : les caractères de la chaine :

```
Alphabet = " abcdefghijklmnopqrstuvwxyz"
```

1) Ecrire une fonction qui vérifie que deux chaines de caractères contiennent les mêmes caractères.
2) Ecrire une fonction qui vérifie que deux chaines de caractères sont des anagrammes l'une de l'autre.
3) Si vous avez utilisé un tri à la question précédente, écrire une fonction plus efficace n'utilisant pas de fonction de tri.

Ex 5 :

1) Écrire une fonction `gene(n)` qui prend en entrée un entier naturel non nul n et retourne une chaîne de n caractères formée aléatoirement, de façon équiprobable, des caractères 'A', 'C', 'G', 'T'.
2) Écrire une fonction `nbAC(g)` qui prend en entrée une chaîne de caractères formée des caractères 'A', 'C', 'G', 'T' et qui retourne le nombre de fois où la séquence 'AC' est présente.

   Par exemple, `nbAC('GAGCACCCTACTTGGCGCGA')` retournera 2.

Ex 6 :

1) Écrire une fonction `somme(f, a, b, N)` qui retourne $\displaystyle\sum_{k=0}^{N-1} f\left(a + k\frac{b-a}{N}\right)$ où f est une fonction, a et b deux réels ($a < b$), et N un entier supérieur à 1.
2) Écrire une fonction prenant en entrée une fonction `f`, deux réels `a`, `b` qui renvoie une valeur approchée de $\displaystyle\int_a^b f(t)\mathrm{d}t$.

   Utiliser cette fonction pour donner une valeur approchée de $\displaystyle\int_0^{+\infty} \exp\left(-t^2\right)\mathrm{d}t$ à l'aide de l'égalité :

$$\int_0^{+\infty} f(t)\,\mathrm{d}t = \int_0^1 \frac{1}{(1-t)^2} f\left(\frac{t}{1-t}\right)\mathrm{d}t.$$

Ex 7 : On s'intéresse dans cet exercice à des listes d'entiers. L'utilisation de `max` ou de `count` est interdite.

1) Écrire une fonction prenant en argument une liste d'entiers `L` et un entier $k \in \mathbb{N}$. Cette fonction renvoie `True` si tous les entiers de cette liste sont compris entre 0 et k et `False` sinon.
   Par exemple, pour $\mathrm{L} = [0, 2, 0]$, la fonction renvoie `True` si $k = 2$ ou $k = 3$ et `False` si $k = 1$.
2) Écrire une fonction de mêmes arguments, où la liste est supposée ne contenir que des entiers compris entre 0 et k. Cette fonction renvoie l'élément le plus fréquent (ou l'un d'entre eux s'il y en a plusieurs).
   Par exemple, pour `L = [0, 4, 0, 1, 4]` et `k = 3`, la fonction renvoie 0 ou 4.

Ex 8 : Soit $n \in \mathbb{N}$ et $(a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$. On considère le polynôme $P(X) = a_0 + a_1X + \ldots + a_nX^n$.

Dans Python, on représente ce polynôme P par la liste de ses coefficients $[a_0, \ldots, a_n]$

1) Écrire une fonction Python `eval` qui prend en argument un polynôme P et un réel a et qui renvoie $P(a)$.
2) Écrire une fonction Python `deriv` qui prend en argument un polynôme P et qui renvoie P'.
3) On considère l'application linéaire f définie sur $\mathbb{R}[X]$ par $f(P)(X) = 2XP(X) - P'(X)$.
   Écrire une fonction Python `f` qui prend en argument un polynôme P et qui renvoie $f(P)$.

Ex 9 :

1) Écrire une fonction `somme_cumul` qui prend en entrée une liste L d'entiers et qui renvoie une liste M des sommes cumulées, c'est à dire une liste M, de même longueur que L, telle que $M[i] = \displaystyle\sum_{k=0}^{i} L[k]$ pour tout indice i de L.
2) On propose à deux joueurs A et F la résolution d'une suite infinie de problèmes numérotés $0, 1, \ldots, n, \ldots$.
   Ils débutent la résolution à l'instant 0 du problème numéro 0 . Dès que l'un des joueurs termine la résolution du problème k il passe au problème $k + 1$.
   Le numéro d'un problème est attribué au joueur qui le résout en premier. En cas de simultanéité de la résolution d'un problème par les deux joueurs celui-ci n'est pas attribué.
   On souhaite écrire une fonction Python `répartition(duréesA, duréesF)` qui, étant donnés les deux listes des durées de résolution des n premiers problèmes par les deux joueurs, renvoie les listes des numéros des problèmes attribués à A et F pour les n premiers problèmes.
   a. `répartition([7,1,2,1,2],[4,2,4,3,1])` renvoie (`[3,4], [0,1]`).
      Quel couple renvoie l'exécution de `répartition([3,2,5,1,1,1],[4,2,2,4,4,2])` ?
   b. Écrire la fonction recherchée.

Ex 10 :

1) Écrire une fonction python, d'arguments d'entrée i et N, qui simule une marche aléatoire sur $\mathbb{Z}$: le marcheur démarre sur l'entier i et à chaque pas, il avance de 1 ou recule de 1 avec probabilité 1/2. La marche s'arrête après le N-ième pas et la fonction renvoie l'entier sur lequel le marcheur s'est arrêté.

   Un plateau de jeu est constitué de $n+1$ cases numérotées de 0 à n avec $n > 1$, les cases 0 et n contenant des valeurs a et b, comme ci-dessous où $n = 10, a = 1$ et $b = 3$:

1										3
0	1	2	3	4	5	6	7	8	9	10

   Pour $i \in \{0, 1, \ldots, n\}$, on considère le jeu J_i suivant : on place un pion dans la case de numéro i et tant que l'on n'est pas dans la case 0 ou dans la case n, on lance une pièce équilibrée : si elle tombe sur face, on se déplace vers la gauche ; sinon, on se déplace vers la droite. On admet que la partie se termine presque sûrement : le gain du joueur est la valeur (a ou b) contenue dans la case finale.
2) Écrire une fonction qui prend en paramètres les valeurs n, a, b et i, qui simule ce jeu et qui renvoie le gain.