Vous pouvez utiliser votre ordinateur pour écrire les fonctions demandées. Toutes ces fonctions seront placées dans un fichier intitulé `NOM_TPNOTE.py` (vous remplacerez NOM par votre nom de famille), que vous m'enverrez par mail à la fin du TP noté : `pierre-alain.reynier@univ-amu.fr`

Je vous conseille de tester les fonctions que vous écrivez au fur et à mesure, dans la partie 'programme principal' de votre fichier. Vous pouvez laisser cette partie dans le fichier que vous m'envoyez, je n'en tiendrai pas compte.

Exercice I.1 *(Structure conditionnelle (2 points))*

Ecrivez une fonction `couleur` qui prend en entrée un entier naturel N et retourne la chaîne de caractères :
— "rouge" si N est un nombre pair strictement plus grand que 10,
— "vert" si N est un nombre impair inférieur ou égal à 25,
— "bleu" sinon.

Exercice I.2 *(Somme des carrés des nombres impairs (5 points : 1+2+2))*

1. Ecrire une fonction `polynome` qui prend en argument un entier n, et retourne la valeur de $\frac{n(4n^2-1)}{3}$.
2. Ecrire une fonction `sci` (somme des carrés des entiers impairs) qui prend en argument un entier n, et retourne la valeur de la somme suivante :

$$S = \sum_{i=1}^{n} (2i-1)^2 = 1^2 + 3^2 + 5^2 + \ldots + (2n-1)^2$$

3. Nous affirmons que la formule introduite à la question 1. permet de calculer la somme introduite à la question 2. Pour cela nous allons comparer les valeurs obtenues pour un entier n compris entre 1 et N. Écrire une fonction `test_formule` prenant en entrée un entier N et retournant `True` si pour tous les entiers n compris entre 1 et N, la formule est correcte, et retournant `False` sinon.

Exercice I.3 *(Fréquence des lettres (4 points : 2+2))*

La fréquence d'un symbole dans une chaîne de caractères est définie comme le nombre d'occurrences de ce symbole dans la chaîne, divisé par la longueur de la chaîne.

Dans cet exercice, on considèrera des chaînes de caractères utilisant uniquement les lettres a, b, c et d, et de longueur au moins 1.

1. Écrire une fonction `compte` prenant en entrée une chaîne de caractères `chaine` et un caractère `char` et retournant le nombre d'occurrences de `char` dans `chaine`.
2. Écrire une fonction `frequence` prenant en entrée une chaîne de caractères `chaine` et retournant une liste de quatre éléments $L = [x_1, x_2, x_3, x_4]$, telle que x_1 est égal à la fréquence de a dans `chaine`, x_2 à la fréquence de b dans `chaine`, ainsi de suite.

Exercice I.4 *(Génération aléatoire (4 points : 2+2))*

On souhaite construire des chaînes de caractères aléatoires. Pour cela, vous utiliserez la librairie `random`.

1. Écrire une fonction `genere_char` qui prend en entrée une liste $L = [n_1, n_2, n_3, n_4]$ de quatre entiers naturels tels que $n_1 + n_2 + n_3 + n_4 = 100$. Elle doit retourner un caractère déterminé de la façon

suivante. Un nombre aléatoire n est tiré dans l'intervalle $[0, 99]$ à l'aide de la fonction `randrange` et le caractère retourné est :

$$\begin{cases} a & \text{si } n < n_1 \\ b & \text{si } n_1 \leqslant n < n_1 + n_2 \\ c & \text{si } n_1 + n_2 \leqslant n < n_1 + n_2 + n_3 \\ d & \text{sinon} \end{cases}$$

2. Écrire une fonction `genere` qui prend en entrée en entier naturel N et une liste $L = [n_1, n_2, n_3, n_4]$ de quatre entiers naturels tels que $n_1 + n_2 + n_3 + n_4 = 100$, et qui retourne une chaîne de caractères composée de N caractères produits à l'aide de la fonction de la question précédente.

Exercice I.5 *(Compression/décompression (5 points : 2+3))*

On souhaite transformer une chaîne de caractères en une autre, de sorte à la compresser. Pour cela, l'idée générale consiste à utiliser une représentation courte pour les caractères très fréquents, et une plus longue pour les caractères peu fréquents.

1. Écrire une fonction `code` prenant en entrée une chaîne de caractères `chaine` composée uniquement des lettres a, b, c et d, et une liste M composée de 4 chaînes de caractères. La fonction `code` retourne la chaîne de caractères obtenue à partir de `chaine` en remplaçant chaque a par la chaîne $M[0]$, chaque b par la chaîne $M[1]$, et ainsi de suite. Par exemple, `code('aba',['0','100','101','110'])` retourne `01000`.
2. On considère la liste $M = ['0','100','101','110']$. On peut démontrer que pour cette liste, le codage proposé dans la question précédente est correct, au sens où toute chaîne de 0 et de 1 possède un unique décodage. Écrire une fonction `decode` qui prend en entrée une chaîne de caractères `codage` composée uniquement de 0 et de 1, et qui retourne la chaîne `chaine` telle que `code(chaine,N)=codage`, si elle existe, et qui retourne `False` sinon. Pour cela, vous pouvez parcourir la chaîne de gauche à droite afin d'identifier la décomposition de la chaîne `codage`. Par exemple, `decode('01100101')` retourne `'adac'`.

Exercice I.6 *(Question bonus : évaluation (2 points))*

On utilise la liste $L = [70, 10, 10, 10]$ pour générer des chaînes de caractères aléatoires composées des lettres a, b, c et d, comme indiqué dans l'exercice 4.

Dans un premier temps, écrivez une fonction pour vérifier, à l'aide des fonctions de l'exercice 3, que les chaînes ainsi produites ont une fréquence convergeant vers $[.7, .1, .1, .1]$. Cette fonction générera une telle chaîne de taille N et affichera ses statistiques de fréquence. Vous pouvez faire grandir la longueur N des chaînes construites pour observer la convergence.

Dans un second temps, écrivez une fonction qui prend en entrée un entier N, construit une chaîne aléatoire `chaine` de longueur N à l'aide de la question 4.2, et retourne le ratio longueur de `code(chaine)` divisé par $2N$. Ceci tend vers 0.8. Expliquez pourquoi (vous placerez vos explications en commentaire dans votre fichier).