

1 Remise dans le bain : tracer une fonction, lecture/écriture de fichier

Question 1 – Lire le fichier `fichierTP03.txt` à la main. Puis en utilisant `loadtxt(fichier)` de `numpy`, lire le fichier. Ce fichier regroupe les composantes x et y d'une fonction. Tracer cette fonction en rouge.

```
file = loadtxt('fichierTP03.txt')
X_file = file[:,0] # extrait la première colonne
Y_file = file[:,1] # extrait la deuxième colonne
plot(X_file, Y_file, color = "red", linewidth = 2)
grid("on")
show() %f nulle pour x=2 et x=5
```

Question 2 – Définir la fonction f puis à l'aide de `linspace` et `plot`, tracer la fonction f sur $[-,10]$ en rouge. Pensez à passer par un vecteur X et $Y = f(X)$. Attention, vous n'aurez pas le droit de faire de boucle. Quelles sont les valeurs de x pour lesquels f est nulle ?

```
def f(x):
    return x**2-7*x+10

X = linspace (-,10,1000)
Y = f(X)

plot(X,Y,color = "blue", linewidth = 2)
grid("on")
show() %f nulle pour x=2 et x=5
```

Question 3 – Bonus: Quelle est la relation entre la fonction du fichier et celle tracée ? Vérifier votre hypothèse en traçant la fonction en question.

```
# La fonction du fichier est l'intégrale de la fonction F
# en intégrant on arrive à :
def F(x):
    return x**3/3-7*x**2/2+10*x

FY = F(X)
plot(X,FY,color="green",linewidth=2)
show() # les deux fonctions sont bien confondues
```

Question 4 – À l'aide de `savetxt` de `numpy`, (`loadtxt` existe aussi), sauvegarder dans un fichier texte vos points X, Y . Vous devez dans un premier temps créer un vecteur avec `zeros` de taille $(n, 2)$.

```
fichier = zeros((1000,2))
fichier[:,0] = X
fichier[:,1] = Y

savetxt("fichier.txt", fichier)
```

2 Cryptage de César

Question 5 – Initialisez les deux alphabets suivants :

```
alphabet = "abcdefghijklmnopqrstuvwxyz"
ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Question 6 – Commencez par créer une fonction `codage_lettre(lettre, decalage)`. Vérifiez que pour `codage_lettre(y, 3)` la fonction renvoie bien `b`.

```
def codage_lettre(lettre, decalage):
    for i in range(26): #parcours les indices des lettres
        if alphabet[i]==lettre: #identifie l'indice de la lettre
            return alphabet[(i+decalage)%26] #retourne la lettre décalée
        elif ALPHABET[i]==lettre: #pareil pour les majuscules
            return ALPHABET[(i+decalage)%26]
        else: # si caractère spéciaux, renvoie le même
            return lettre

print(codage_lettre("y", 3))
```

Question 7 – Faire une fonction `codage_cesar(mot, decalage)` qui appellera la fonction précédente pour renvoyer un mot codé. Vérifiez votre fonction.

```
def codage_cesar(mot, decalage):
    mot_code=''
    for lettre in mot:
        mot_code+=codage_lettre(lettre, decalage)
    return mot_code

print(codage_cesar("mot", 3))
```

Question 8 – Créer maintenant une fonction `decodage_cesar(mot, decalage)` faisant la manipulation inverse.

```
def decodage_cesar(mot, decalage):
    return codage_cesar(mot, -decalage)

print(decodage_cesar(codage_cesar("mot", 3), 3))
```

Question 9 – Maintenant, vous allez créer une fonction `lecture(fichier)` qui va permettre de lire le fichier que je vous ai envoyé.

```
def lecture(fichier):
    file = open(fichier, "r")
    for line in file:
        line= line.rstrip('\n')
    file.close()
    return line
```

Question 10 – En essayant tous les décalages possibles (26 maximums), affichez toutes les tentatives de décodages de mon message. Quelle est la clé de décodage? Quel est mon message?

```
phrase= lecture("phrase.txt")
for i in range(1, 27):
    print(decodage_cesar(phrase, i))
```

Question 11 – Bonus : En utilisant `open(fichier, 'w')` pour écrire un fichier, écrivez-moi votre message codé. Envoyez celui-ci à mon adresse : ronan.dupont@umontpellier.fr.

```
phrase_pour_ronan = "Merci pour ce super TP ahah"
phrase_pour_ronan_code = codage_cesar(phrase_pour_ronan, 7)

fichier_ronan = open("phrase_code.txt", 'w')
fichier_ronan.write(phrase_pour_ronan_code)
fichier_ronan.close()
```