

## DM1 - CORRECTION

### Exercice 0.1. Parcours de liste, chaîne et dictionnaire

1. Créer un code utilisant un dictionnaire **comptage\_lettre** permettant d'afficher le nombre de chaque lettre dans une chaîne de caractères test.

On utilisera un alphabet `alphabet = 'abcdefghijklmnopqrstuvwxy'`

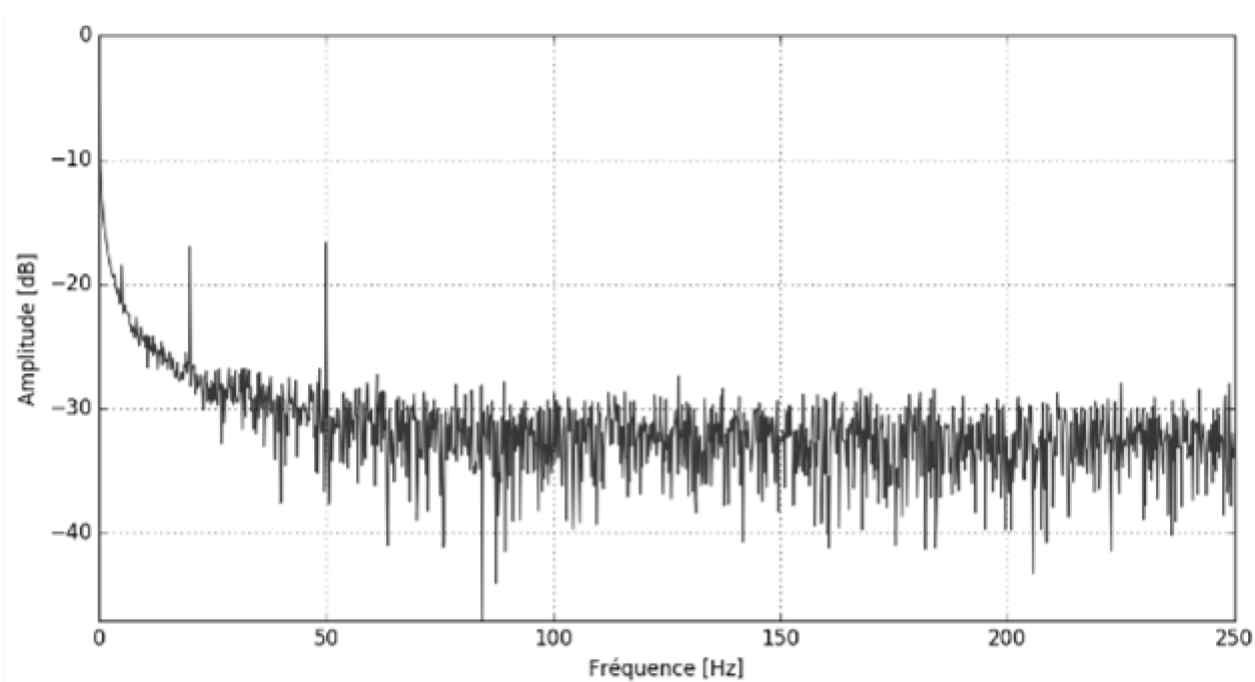
Exemple : pour `test='ilyauradelinfoauconcours'` : 'a' :3, 'b' :0 , 'c' :2, 'd' :1, etc...

2. Créer un code utilisant une liste **comptage\_chiffre** permettant d'afficher le nombre de chaque chiffre dans une chaîne de caractères test 2 (exemple : pour `test2= '3147216839371'` : '1' :3, '2' :2, '3' :3, etc...

```
alphabet = 'abcdefghijklmnopqrstuvwxy'
test = 'ilyauradelinfoauconcours'
comptage_lettre={}
for lettre in alphabet:
    comptage_lettre[lettre]=0
for lettre in test:
    comptage_lettre[lettre]+=1
for lettre in comptage_lettre:
    print(lettre, ': ', comptage_lettre[lettre])
test2= '3147216839371'
```

```
comptage_chiffre= [0 for n in range(10)]
for i in range(len(test2)):
    comptage_chiffre[int(test2[i])] +=1
for i in range(10):
    print(i, ': ', comptage_chiffre[i])
```

### Exercice 0.2. Le calcul du spectre du signal au moyen de la transformée de Fourier discrète a permis d'obtenir le tracé de la figure ci dessous :



Le bruit se manifestant à toutes les fréquences, il est raisonnable de le modéliser par un bruit blanc gaussien.

Afin de tester le filtre qui sera utilisé dans l'application, nous allons utiliser un signal simulé représentatif auquel on ajoutera ensuite un bruit gaussien.

Le signal représentatif est déterminé à partir d'une analyse temporelle du signal mesuré d'une part et d'une analyse spectrale d'autre part.

Il est la somme :

- d'une réponse indicielle d'un système du premier ordre de constante de temps  $\tau = 500$  ms et de valeur finale  $x_f = 5320$ , soit  $s(t) = x_f (1 - e^{(-t/\tau)})$  ;
- de trois signaux sinusoïdaux modélisant les harmoniques du signal réel :
  - de fréquences respectives :  $f_1 = 10$  Hz,  $f_2 = 20$  Hz,  $f_3 = 50$  Hz ;
  - d'amplitudes respectives :  $a_1 = 200$ ,  $a_2 = 200$ ,  $a_3 = 200$  ;
  - de déphasage :  $\varphi_1 = 0$ ,  $\varphi_2 = \frac{\pi}{3}$ ,  $\varphi_3 = \frac{\pi}{5}$ , soit pour le premier  $a_1 \sin(2 \cdot \pi \cdot f_1 \cdot t + \varphi_1)$ .

1. a) Écrire la fonction `signal()` prenant en argument la variable  $t$  et renvoyant la valeur du signal  $x_1(t)$ .
2. Finalement, pour construire le signal modélisant la mesure issue du capteur, on ajoute à  $x_1$  un signal appelé bruit blanc.

On parle de bruit gaussien lorsque la densité de probabilité de cette variable est la loi gaussienne (ou loi normale).

En choisissant une loi gaussienne centrée, la densité de probabilité  $p(x)$  est : Les signaux délivrés par les instruments de mesure comportent un bruit qui peut

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}$$

être généralement considéré comme gaussien. Cette hypothèse permet de définir l'écart-type  $\sigma$  comme l'incertitude type due aux erreurs aléatoires.

On dispose d'un générateur de nombre pseudo-aléatoire délivrant des nombres dans l'intervalle  $[0,1]$  avec une densité de probabilité uniforme.

La génération de nombres aléatoires vérifiant la distribution gaussienne se fait alors en utilisant le théorème suivant :

**Si  $U_1$  et  $U_2$  sont deux variables aléatoires uniformes sur l'intervalle  $[0,1]$  alors  $\sigma\sqrt{-2\ln(U_1)}\cos(2\pi U_2)$  est une variable aléatoire vérifiant la loi gaussienne centrée.**

Rédiger une fonction `bruitGauss()`, prenant en argument l'écart-type  $\sigma$  et renvoyant un nombre aléatoire vérifiant la distribution gaussienne.

```
def signal(t) :
    return ( 200.0*math.sin(10*math.pi*t)\ \
            +200.0*math.sin(20*2*math.pi*t+math.pi/3)\ \
            +200.0*math.sin(50*2*math.pi*t+math.pi/5)\ \
            +5320.0*(1-math.exp(-t/0.5)) )
def bruitGauss(sigma):
    U1 = random.random()
    U2 = random.random()
    return ( sigma*math.sqrt(-2*math.log(U1))*math.cos(2*math.pi*U2) )
```

**Exercice 0.3.** On s'intéresse à la gestion d'une base de données nommée *Personnel* et sera constituée de deux tables intitulées *Employes* et *ListeCatégories*.

- La clé primaire de la table *Employes* est l'email(définie sans son extension@cpp.com).
- site est la liste des sites où l'employé est autorisé à entrer en plus de son site d'origine.
- id est la clé primaire de la table *ListeCatégories*

L'équipe de direction souhaite avoir certaines informations au sujet des employés de l'entreprise.

Table "Employes"

id	nom	prenom	email	age	code	site	code_categorie
1	Genereux	Alain	alain.genereux	47	1AAACDEF	0	1
2	Tanguy	Alain	alain.tanguy	22	1BBACABE	10	8
3	Smith	Alan	alan.smith	47	5BCABCAE	5	2
4	Lefoll	Claude	claud.lefoll	28	2EEABECC	24	1
5	Herberts	Dany	dany.herberts	58	1EEEEABC	0	1
6	Korbs	Eva	eva.korbs	33	2FEAFEAB	9	5
7	Niels	Edwin	edwin.niels	24	2EFDDACA	28	4
8	Joly	Emilie	emilie.joly	25	3FAFABEF	19	4
9	Esteban	Flore	flore.esteban	20	4BECEBAA	12	2
10	Serin	Jacques	jacques.serin	22	2GBBCEAE	21	6
11	Clerc	Jerome	jerome.clerc	29	3DDAABBC	3	1
12	Brown	Katia	katia.brown	46	5AFBECCA	3	1
13	Forbs	Laura	laura.forbs	18	2CBAAEAC	0	1
14	Phil	Marc	marc.phil	33	4BCDABCD	17	7
15	Auzas	Michel	michel.auzas	32	5FECDAEA	15	7
16	Kotta	Michelle	michelle.kotta	27	2ABEEABC	9	7
17	Lambert	Pierre	pierre.lambert	35	3DABGEFC	10	2
18	Klader	Sylvie	sylvie.klader	32	2EAEAEAB	20	1
19	Durand	Sylvie	sylvie.durand	31	1CBBCAEA	16	6
20	Olm	Tatiana	tatiana.olm	25	4EFEFACB	19	3

Table "ListeCategories"

id	categorie
1	ingénieur
2	secrétaire
3	comptable
4	technicien
5	opérateur
6	administratif
7	chef de projet
8	responsable de département

1. Écrire en SQL la requête 1 suivante donnée en algèbre relationnelle :

$$\pi_{nom, prenom, age} (\sigma_{age > 50} (Employes))$$

Que réalise cette requête ?

2. Écrire en SQL la requête 2 donnant comme résultat le nom et la catégorie des personnes de l'entreprise ayant au moins 20 ans.  
Les noms seront classés par ordre alphabétique.
3. Écrire en SQL la requête 3 donnant comme résultat la liste des âges des employés avec comme information associée le nombre d'employés ayant le même âge.  
On demande que cette liste soit décroissante par rapport au nombre de personnes ayant le même âge.  
Par exemple, dans la table Employes, il y a 2 personnes qui ont 22 ans.
4. Écrire en SQL la requête donnant l'email des chefs de projets autorisés à entrer sur le site 3.

1. SELECT nom, prenom, age FROM Employes WHERE age > 50  
Cette requête renvoie les noms, prénoms et âge des employés âgés de strictement plus de 50 ans

2. 

```
SELECT Employes.nom, ListeCategories.categorie
FROM Employes JOIN ListeCategories
ON ListeCategories.id = Employes.code_categorie
WHERE Employes.age >= 20
ORDER BY Employes.nom ASC
```
3. 

```
SELECT age, COUNT(*) as nb
FROM Employes GROUP BY age ORDER BY nb DESC
```
4. 

```
SELECT email FROM Employes JOIN ListeCategories
ON ListeCategories.id = Employes.code WHERE Employes.site=3 AND
ListeCategories.categorie='chef de projet'
```

**Exercice 0.4.** *Écrire un script permettant de résoudre l'équation  $\cos(x) = 0$  sur  $[0; \pi]$ .*

*On utilisera une méthode dichotomique et on donnera une valeur approchée de la solution à  $10^{-2}$  près.*

```
from math import *
def zero_par_dichotomie(f, a, b, epsilon):
    while (b-a) > epsilon:
        milieu=(b+a)/2
        if f(milieu)*f(a)>0:
            a,b=milieu,b
        else:
            a,b=a,milieu
    return milieu

def f(x):
    return (cos(x))

print(zero_par_dichotomie(f,0,pi,0.01))
```