Exercice 1. On suppose qu'on recoit un signal dont les frequences réelles qui nous interessent sont comprises entre 0 et 20Hertzs, avec une précision de 10Hertzs.
a)Quelles sont les frequences réelles et les fréquences complexes correspondantes. Les fréquences réelles sont 0, 10, et 20 Hertzs. Les fréquences complexes sont -20,-10,0,10 et 20
b)Tous les combiens faut il échantillonner le signal ? Indication : on regarde la periode et on la coupe en n morceaux, ou $n =$ le nombre de fréquences. Ici, les frequences sont multiples de 10Hertzs. La periode commune a toutes les exponentielles de la transformée de Fourier est 0.1 s. Comme on a 5 fréquences, on échantillonne tous les 0.1/5=0.02 secondes.
c)Verfifier la formule $Nf = e$, ou $N = nbd'echantillons$, $f = resolution\ frequentielle$, $e = frequenced'echantillonnage$. Ici $N = nb\ d'echantillons = 5$, $f = 10$, on echantillonne toutes les 0.02 secondes, donc 50 fois par secondes, ie. $e = 50$.
d)On suppose que les valeurs reçues sont $s = [0, 1, 2, 3, 0, 1, 2, 3, ...0, 1, 2, 3]$ (40 valeurs en tout). Trouver l'amplitude pour chaque fréquence pendant les premiers instants du signal. La fonction cherchée s'écrit sous la forme $s = ae^{2i\pi.(-20)t} + be^{2i\pi(-10)t} + c + de^{2i\pi.10t} + ee^{2i\pi 20t}$
e)Tracer le graphe à partir des coefficients pour vérifier. Il faut faire une transformée de Fourier mathématiquement. Informatiquement : multiplier la colonne des valeurs par la matrice de transformee de Fourier. Ou uiliser le module fft de numpy/scipy. Attention aux constantes de normalisations !
f)Trouver l'amplitude de chaque fréquence pendant les derniers instants du signal.
g)Trouver l'amplitude de chaque fréquence pendant toute la duree du signal.
h)Expliquer comment trouver les frequences d'un signal musical enregistré sur une minute à 44.1kHz.

Exercice 2. A partir de la formule fondamentale $Nf = e$, et le fait que notre audition va jusque 20 kHz au maximum, expliquer pourquoi l'ordre de grandeur du CD qui echantillonne a 44,1kHz est raisonnable. On prend N donnees temporelles et on fait une transformee de Fourier. Pour fixer les ordres de grandeur, je prends $N = 101$. Pour N donnees temporelles, j'ai N frequences en face. Pour $N = 101$, on 50 frequences positives, 50 frequences negatives, et une frequence nulle. On va avoir un pas frequentiel donné par la formule $Nf = e$, ici $101.f = 44100$. Donc $f = 44100/101$. Les frequences qu'on obtient sont [-50f,...,-2f, -f,0,f,2f,3f,...50f]. La frequence la plus aigue, c'est $50f = 50 * 44100/101 \simeq 22000$ Si on remplace 101 par $2N + 1$, on trouve la frequence maximale de la transformée de Fourier qui est $\frac{N}{2N+1}44100 \simeq \frac{44100}{2}$.

- La frequence maximale consideree correspond a la frequence d'echantillonnage /2 environ (pour le cd 44100 echantillonnage, frequence environ 22050 qui correspond a la limite de l'autdition humaine.
- Plus on prend un grand nombre de donnees pour la transformee de Fourier, plus la resolution frequentielle est fine.

```
import numpy as np
from numpy.fft import fft, ifft
import matplotlib.pyplot as plt
from math import pi
s=[0,1,2,3]*10 # liste de longueur 40 [0,1,2,3,0,1,2,3,0,1,...]
valFonc=s[0:5] #[0,1,2,3,0]
coeffs=1/5*fft(valFonc)# le 1/5 devant est pour etre raccord  avec les
#conventions du cours
#car numpy/scipy n'ont pas le meme def de fft.
print("module des coefficients", np.abs(coeffs))
frequences=np.array([0,10,20,-20,-10])# dans Fourier les frequences
#commencent toujours en 0. Et donc on ne met pas les fr\'equences dans l'ordre
```

```
#[-20,-10,0,10,20] mais dans l'ordre [0,10,20, -20,-10]
print(coeffs)

def maFonc(t):
    freqsx=1j*t*2*pi*frequences #mathematiquement les exposants des exponentielles : [0, 20i pi,4
    valeurs=np.exp(freqsx) # valeurs des fonctions exponentielles
    values=np.dot(coeffs,valeurs)
    #print("imaglnaires",np.imag(values))
    return np.real(values)
t = np.linspace(0, .1, 100)
y=[maFonc(x) for x in t]

print("check Valeurs",maFonc(0),maFonc(1./50),maFonc(2./50),maFonc(3./50),maFonc(4./50))
#plt.plot(t,y)
#plt.show()
```

Applications de la transformée de Fourier discrete:

- resampler: faire un fourier, puis calculer le signal en d'autres points.
- Faire un accordeur de guitare: faire Fourier, puis prendre la frequence ayant le coefficient le plus haut.
- Faire un mini-equalizer : Faire fourier, puis modifier la puissance de chaque frequence.
- FIltre passe haut, passe bas.
- Lisser une fonction. Exemple de la meteo. On a 365 temperatures sur l'annee. On fait une transformee de fourier. On obtient 365 frequences. Les basses frequences correspondent aux evolutions lentes (mois, semestre,....). Les hautes frequences aux evolutions rapides (journalieres...). On met un certain nombre de hautes frequences a 0. Et on fait une transformee de Fourier inverse. On obtient un nouveau diagramme de températures, mais beaucoup plus lisse visuellement parce que vous avez avez filtré le signal et conservé uniquement des basses fréquences (filtre passe bas).