Experiment No.4: Discrete Fourier Transform

In [ ]:
```
'''
Name: Om Kadam
Roll No: 45
Class: TE-EXTC A
Division: A
Year Of Study: TE
Branch: EXTC
Date: 05/08/2023
Time: 14:00
'''
```

Problem Statement: Write a code to implement DFT & IDFT using Formula Method.

In [2]:
```
#Importing Python Libraries
import numpy as np
import matplotlib.pyplot as plt
```

In [64]:
```
#Getting user-defined inputs
x = eval(input("Enter input sequence x[n]= "))
L = len(x)
print("Length of x[n] = ",L)
N = 4
```

```
Enter input sequence x[n]= [1,2,3,4,5]
Length of x[n] =  5
```

In [65]:
```
#Callable Function for DFT
X = np.zeros(N,complex)
def DFT_TEA (x,N):
    for k in range(N):
        for n in range(N):
            X[k]+= x[n] * np.exp((-2j * np.pi * n * k)/N)
    return(X)
```

In [66]:
```
#Calling DFT Function
X = np.zeros(N,complex)
if N < L:
    print("DFT cannot be computed")
else:
    X = np.round(DFT_TEA(x,N),decimals = 2)

    print(X)
```

```
DFT cannot be computed
```

In [68]:
```
#Getting user-defined inputs
X = eval(input("Enter input sequence X[k]= "))
L = len(X)
print("Length of X[k] = ",L)
N = 4
```

```
Enter input sequence X[k]= [10,-2-2j,-2,-2+2j]
Length of X[k] =  4
```

In [73]:
```
#Callable Function for IDFT
x = np.zeros(N,complex)
```

```
def IDFT_TEA (X,N):
    for n in range(N):
        for k in range(N):
            x[n]+= X[k] * np.exp((2j * np.pi * n * k)/N)/N
    return(x)
```

In [74]:

```
#Calling IDFT Function
x = np.zeros(N,complex)
if N < L:
    print("IDFT cannot be computed")
else:
    x = np.round(IDFT_TEA(X,N),decimals = 2)

    print(x)
```

```
[1.+0.j 4.-0.j 3.-0.j 2.+0.j]
```

In [78]:

```
#Verification with inbuilt function
X1 = [10,-2-2j,-2,-2+2j]
print(np.fft.ifft(X1))
```

```
[1.+0.j 4.+0.j 3.+0.j 2.+0.j]
```