Machine Learning for Signal Processing and Pattern Classification.

DISCRETE FOURIER TRANSFORM (DFT)

SIMRAN

1.

AIM:

For the given samples of the signal x = [1 2 3 4 5 6 7 8], compute the DFT coefficients using built-in ‘fft’ command in Matlab.

CODE:

```
X = [1 2 3 4 5 6 7 8]
Y = fft(X)'  % finding DFT coefficients using built-in command ‘fft’
```

DFT coefficients were found using built-in command ‘fft’

OUTPUT:

```
X =

    1    2    3    4    5    6    7    8

Y =

  36.0000 + 0.0000i

  -4.0000 - 9.6569i

  -4.0000 - 4.0000i

  -4.0000 - 1.6569i

  -4.0000 + 0.0000i

  -4.0000 + 1.6569i

  -4.0000 + 4.0000i

  -4.0000 + 9.6569i
```

2.

AIM:

To compute the DFT matrix for the length of the signal given in Q.No.1 (Note: Generalize the length based on the given input signal. Don’t hard code as ‘8’ as per the given signal samples) using the built-in command ‘dftmtx’. Compute the DFT coefficients using the generated matrix. Verify the answer with the answer obtained for Q.No.1

CODE:

```
X = [1 2 3 4 5 6 7 8]
N = length(X);
X1= dftmtx(N)*X'  % finding the DFT coefficients using built-in ‘dftmtx’
```

Dftmtx value is multiplied with the signal to get the dft coefficients.
The output coefficient values are same as the values when fft is used.

OUTPUT:

```
X =

   1    2    3    4    5    6    7    8

X1 =

  36.0000 + 0.0000i

  -4.0000 + 9.6569i

  -4.0000 + 4.0000i

  -4.0000 + 1.6569i

  -4.0000 + 0.0000i

  -4.0000 - 1.6569i

  -4.0000 - 4.0000i

  -4.0000 - 9.6569i
```

3.

Aim:

To write our own code to generate the DFT matrix for any length of the signal, which must be the power of 2. Use the following mathematical expression as the reference to generate the matrix. Use the generated matrix to compute the DFT coefficients and verify the answer with the answers obtained using the built-in command for Q. No 1 and 2.

CODE:

```
x=[2 4 8 16 32];
if(log2(x))
N=length(x);  % N is the length of signal
n=0:N-1;   % n is the number of columns
theta=2*pi*n/N;  % theta value are taken
k=(0:N-1)';  % k is the number of rows
W=exp(-i*k*theta)*x'  % coefficients of DFT
w=dftmtx(N)*x' % built-in command for dft coefficients
 X=fft(x)'  % built-in command for fft
else
disp('Not a valid signal')
end
```

Using the formula, dft coefficients were calculated and verified using the built-in functions.

OUTPUT

FIRST TIME

```
x =

    2    4    8   16   32

W =

  62.0000 + 0.0000i

  -6.2918 +31.3319i
```

```
 -19.7082 + 8.8495i

 -19.7082 - 8.8495i

  -6.2918 -31.3319i

w =

  62.0000 + 0.0000i

  -6.2918 +31.3319i

 -19.7082 + 8.8495i

 -19.7082 - 8.8495i

  -6.2918 -31.3319i

X =

  62.0000 + 0.0000i

  -6.2918 -31.3319i

 -19.7082 - 8.8495i

 -19.7082 + 8.8495i

  -6.2918 +31.3319i
```

SECOND TIME (not valid signal – changed the x value in the code)

```
x =

     1     2     3     4     5     6     7     8

Not a valid signal
```