Experiment No 4

i) To find the 2N point DFT using N point DFT

```
close all;
clear all;
v=input('Enter the Input sequence v[n]');
N=length(v)/2

for n=0:N-1
    g(n+1)=v((2*n)+1);
    h(n+1)=v((2*n)+2);
end

[G,H]=myN_Point(g,h);


for k=0:(2*N)-1
    w(k+1)=exp((-i*pi*k)/N);
end

m=0;
for k=1:2
    for n=0:N-1
        V(m+n+1)=G(n+1)+(w(m+n+1)*H(n+1));
    end
    m=4;
end

disp('2N -Point DFT using N point DFT is V(K)')
disp(V)
****************************************

function[G,H]=myN_Point(g,h)
N=length(g);

%x[n]= g[n]+jh[n]
for i=0:N-1
    x(i+1)=g(i+1)+h(i+1)*j;
end

%Finding DFT of x[n] i.e X[K]
X =mydftusingfft(x,N);

%Finding Conjugate of X(K) i.e X*(K)
Z=conj(X);

%X*[(N-K)N]
```

```
%n=0:N-1
%Z(mod(-n,N)+1)
for k=0:N-1
    n=N-k;
    if(n==N)
        y(k+1)=Z(k+1);
    else
        y(k+1)=Z(n+1);
    end
end

G=zeros(1,N);
H=zeros(1,N);

for k=1:N
    G(k)=(1/2)*(X(k)+y(k));
    H(k)=(X(k)-y(k))/(2*j)
end
end

*********************************************************
**
function y=mydftusingfft(x,N)

%to alter the input sequence is X[0] x[1] x[3]

x=bitrevorder(x);


% To find n from 2^n i.e No of stages
%p=nextpow2(N);
M=log2(N);
h=1;


for stage = 1:M
    for index = 0:(2^stage):N-1
        for n=0:(h-1)
            pos =n+index+1;
            pow= (2^(M-stage)*n);
            w=exp((-i)*(2*pi)*pow/N);
            a=x(pos)+x(pos+h).*w;
            b=x(pos)-x(pos+h).*w;
            x(pos)=a;
            x(pos+h)=b;
        end
    end
    h=2*h;
```

```
end
y=x;
end
```

Result

```
Enter the Input sequence v[n] [1 2 2 2 0 1 1 1]

N =

     4


H =

     6     0     0     0


H =

   6.0000 + 0.0000i   1.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i


H =

   6.0000 + 0.0000i   1.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i


H =

   6.0000 + 0.0000i   1.0000 - 1.0000i   0.0000 + 0.0000i   1.0000 + 1.0000i

2N -Point DFT using N point DFT is V(K)
  10.0000 + 0.0000i   1.0000 - 2.4142i  -2.0000 + 0.0000i   1.0000 - 0.4142i  -2.0000 -
0.0000i   1.0000 + 0.4142i  -2.0000 + 0.0000i   1.0000 + 2.4142i
```

ii)To find the N point DFT of two sequences using single N point DFT(Using FFT algorithm to find DFT)

```
close all;
clear all;
```

```
g=input('Enter the first  sequence g[n]');
h= input('Enter the second sequence h[n]');
N=length(g)

for i=0:N-1
    x(i+1)=g(i+1)+h(i+1)*j;
end
disp(x)


%Finding DFT of x[n] i.e X[K]
X =mydftusingfft(x,N)

%Finding Conjugate of X(K) i.e X*(K)
Z=conj(X)

%X*[(N-K)N]
%n=0:N-1
%Z(mod(-n,N)+1)
for k=0:N-1
    n=N-k;
    if(n==N)
        y(k+1)=Z(k+1);
    else
        y(k+1)=Z(n+1);
    end
end
disp(y)

G=zeros(1,N);
H=zeros(1,N);

for k=1:N
    G(k)=(1/2)*(X(k)+y(k));
    H(k)=(X(k)-y(k))/(2*j);
end

disp('N Point DFT of two sequences using N point DFT')
disp(G)
disp(H)
```

```
function y=mydftusingfft(x,N)

%to alter the input sequence is X[0] x[1] x[3]

x=bitrevorder(x);
```

```
% To find n from 2^n i.e No of stages
%p=nextpow2(N);
M=log2(N);
h=1;


for stage = 1:M
    for index = 0:(2^stage):N-1
        for n=0:(h-1)
            pos =n+index+1;
            pow= (2^(M-stage)*n);
            w=exp((-i)*(2*pi)*pow/N);
            a=x(pos)+x(pos+h).*w;
            b=x(pos)-x(pos+h).*w;
            x(pos)=a;
            x(pos+h)=b;
        end
    end
    h=2*h;
end
y=x;
end
```

Result:

```
Enter the first  sequence g[n] [1 2 0 1]
Enter the second sequence h[n] [ 2 2 1 1]

N =

     4

   1.0000 + 2.0000i   2.0000 + 2.0000i   0.0000 + 1.0000i   1.0000 + 1.0000i


X =

   4.0000 + 6.0000i   2.0000 + 0.0000i  -2.0000 + 0.0000i   0.0000 + 2.0000i


Z =

   4.0000 - 6.0000i   2.0000 - 0.0000i  -2.0000 + 0.0000i   0.0000 - 2.0000i

   4.0000 - 6.0000i   0.0000 - 2.0000i  -2.0000 + 0.0000i   2.0000 - 0.0000i
```

N Point DFT of two sequences using N point DFT

```
4.0000 + 0.0000i   1.0000 - 1.0000i  -2.0000 + 0.0000i   1.0000 + 1.0000i
6.0000 + 0.0000i   1.0000 - 1.0000i   0.0000 + 0.0000i   1.0000 + 1.0000i
```