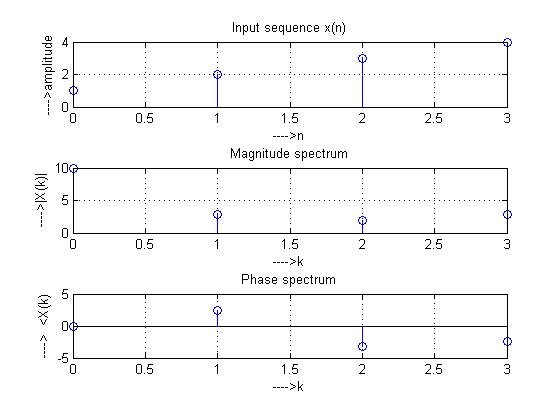
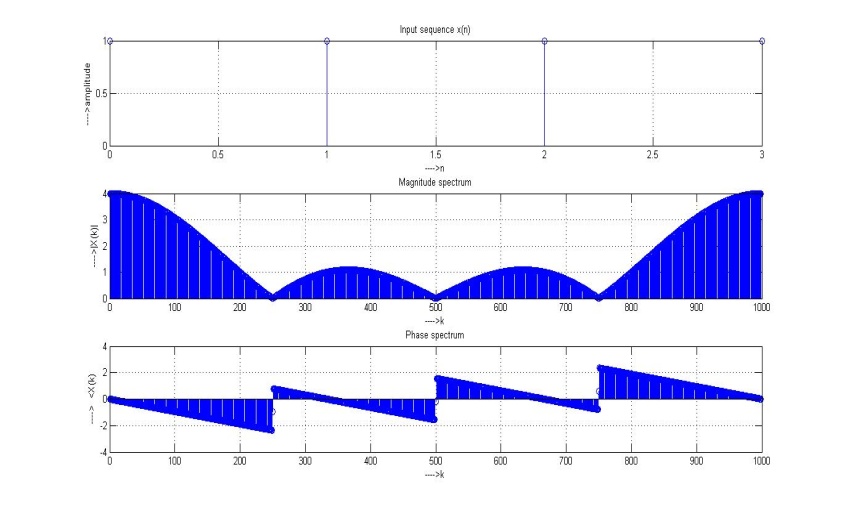
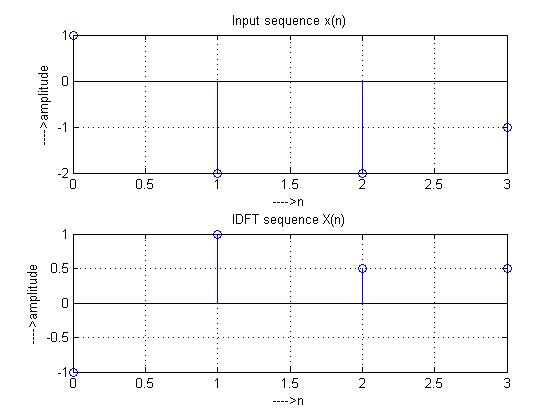
|  |
| --- |
| %Computation of N Point DFT a sequence  clc;  clear all;  close all;    x1=input('Enter the sequence x(n): ');  N=input('Enter the length of DFT: ');  x=[x1,zeros(1,N-length(x1))]  k=0:N-1  n=k  powers=n.'\*k;  disp('DFT matrix for given is');  WNkn=(exp(-j\*2\*pi/N)).^powers  disp('The N point DFT of the given sequence is')  X=WNkn\*x.' % x is sequence after zero padding  disp('The N point DFT of the given sequence using inbuilt function is')  X1=fft(x1,N) % for inbuilt use original sequence  disp('Magnitude spectrum:')  X\_mag=abs(X)  disp('Phase spectrum:')  X\_ang=angle(X)    figure  subplot(3,1,1);  stem(0:length(x1)-1,x1); grid on;  xlabel('---->n');  ylabel('---->amplitude');  title('Input sequence x(n)')    subplot(3,1,2);  stem(0:length(X\_mag)-1,X\_mag); grid on;  xlabel('---->k');  ylabel('---->|X(k)|');  title('Magnitude spectrum')    subplot(3,1,3);  stem(0:length(X\_ang)-1,X\_ang); grid on;  xlabel('---->k');  ylabel('----> <X(k)');  title('Phase spectrum') |

|  |
| --- |
| OUTPUT:  DFT matrix for given is  WNkn =  1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i  1.0000 + 0.0000i 0.0000 - 1.0000i -1.0000 - 0.0000i -0.0000 + 1.0000i  1.0000 + 0.0000i -1.0000 - 0.0000i 1.0000 + 0.0000i -1.0000 - 0.0000i  1.0000 + 0.0000i -0.0000 + 1.0000i -1.0000 - 0.0000i 0.0000 - 1.0000i  The N point DFT of the given sequence isX =  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 - 2.0000i  The N point DFT of the given sequence using inbuilt function is  X1 =10.0000 + 0.0000i -2.0000 + 2.0000i -2.0000 + 0.0000i -2.0000 - 2.0000i  Magnitude spectrum:  X\_mag =  10.0000  2.8284  2.0000  2.8284  Phase spectrum:  X\_ang =  0  2.3562  -3.1416  -2.3562 |





|  |
| --- |
| %Computation of N Point IDFT a sequence  clc;  clear all;  close all;    x1=input('Enter the sequence X(n): ');  N=input('Enter the length of IDFT: ');  x=[x1,zeros(1,N-length(x1))]  k=0:N-1  n=k  powers=n.'\*k;  disp('IDFT matrix for given is');  WNkn=(exp(j\*2\*pi/N)).^powers  disp('The N point IDFT of the given sequence is')  X=1/N\*WNkn\*x.' % x is sequence after zero padding  disp('The N point IDFT of the given sequence using inbuilt function is')  X1=ifft(x1,N) % for inbuilt use original sequence      figure  subplot(2,1,1);  stem(0:length(x1)-1,x1);  grid on;  xlabel('---->n');  ylabel('---->amplitude');  title('Input sequence x(n)')    subplot(2,1,2);  stem(0:length(X1)-1,X1);  grid on;  xlabel('---->n');  ylabel('---->amplitude');  title('IDFT sequence X(n)') |
| OUTPUT:  IDFT matrix for given is  WNkn =  1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i  1.0000 + 0.0000i 0.0000 + 1.0000i -1.0000 + 0.0000i -0.0000 - 1.0000i  1.0000 + 0.0000i -1.0000 + 0.0000i 1.0000 - 0.0000i -1.0000 + 0.0000i  1.0000 + 0.0000i -0.0000 - 1.0000i -1.0000 + 0.0000i 0.0000 + 1.0000i  The N point IDFT of the given sequence is  X =  -1.0000 + 0.0000i  1.0000 - 0.5000i  0.5000 + 0.5000i  0.5000 - 0.0000i  The N point IDFT of the given sequence using inbuilt function is  X1 =  -1.0000 + 0.0000i 1.0000 - 0.5000i 0.5000 + 0.5000i 0.5000 + 0.0000i |



|  |
| --- |
| //C Program To find N point DFT of a given sequence  **#define** CHIP\_6713 1  **#include** "dsk6713.h"  **#include** "dsk6713\_aic23.h"  **#include** "stdio.h"  **#include** "stdlib.h"  **#include** "math.h"  **#define** pi 3.141592653589793  **float** \*x, \*XReal, \*XImag, \*XMag, \*XPhase;  **void** **main**()  {  **int** i, n , k, N, lx;  //Reading length of input sequence  **printf**("\nEnter the length of input sequence\n");  **scanf**("%d",&lx);  //Read the value of N  **printf**("\nEnter the N point DFT desired\n");  **scanf**("%d",&N);  x=(**float**\*)**malloc**(N\***sizeof**(**float**));  //Dynamic memory allocation  XReal=(**float**\*)**malloc**(N\***sizeof**(**float**));  XImag=(**float**\*)**malloc**(N\***sizeof**(**float**));  XMag=(**float**\*)**malloc**(N\***sizeof**(**float**));  XPhase=(**float**\*)**malloc**(N\***sizeof**(**float**));  //Reading input sequence  **printf**("\nEnter the input sequence \n");  **for**(i=0;i<lx;i++)  **scanf**("\n%f\n",&x[i]);  //padding of zeros to input sequence  **for**(i=lx;i<N;i++)  x[i]=0;  //printing input sequence after padding zeros  **for**(i=0;i<N;i++)  **printf**("\n%f\n",x[i]);  //Computing DFT  **for**(k=0;k<N;k++)  {  XReal[k]=0;  XImag[k]=0;  **for**(n=0;n<N;n++)  {  XReal[k]+=x[n]\***cos**(2\*pi/N\*k\*n); //real part  XImag[k]-=x[n]\***sin**(2\*pi/N\*k\*n); //imaginary part  }  }  //Printing output  **printf**("\nThe DFT computed is\n");  **for**(k=0;k<n;k++)  **printf**("(%f)+1j\*(%f)\n",XReal[k],XImag[k]);  **for**(k=0;k<n;k++)  {  XMag[k]=**sqrt**(**pow**(XReal[k],2)+**pow**(XImag[k],2)); //magnitude  XPhase[k]=**atan2**(XImag[k],XReal[k]); //phase  }  //Printing phase and Magnitude values  **printf**("The magnitude and phase response of DFT are\n");  **for**(k=0;k<N;k++)  **printf**("XMag[%d]=%f\tXPhase[%d]=%f\n",k,XMag[k],k,XPhase[k]);  } //end of main |

//C Program To find N point DFT of a given sequence

**#define** CHIP\_6713 1

**#include** "dsk6713.h"

**#include** "dsk6713\_aic23.h"

**#include** "stdio.h"

**#include** "stdlib.h"

**#include** "math.h"

**#define** pi 3.1415

**void** **main**(**void**){

**int** n,k,z,q,i,N,E,R,W;

**float** x[100],XImag[100],XReal[100],XMag[100],XPhase[100];

**printf**("enter the lenght of x");

**scanf**("%d",&n);

**printf**("enter the elements of x");

**for** ( i=0;i<n;i++){

**scanf**("%f",x[i]);

}

**printf**("enter the lenght of DFT");

**scanf**("%d",&N);

**if**(N>n) {

**for** (k=n;k<N;k++){

x[k]=0;

}

}

**printf**("printing the zero padded seq");

**for** (z=0;z<=N;z++)

{

**printf**("/t%f",x[z]);

}

// performing dft by splitting real and imaginary

//Computing DFT

**for**(q=0;q<N;q++)

{

XReal[q]=0;

XImag[q]=0;

**for**(n=0;n<N;n++)

{

XReal[q]+=x[n]\***cos**(2\*pi/N\*q\*n); //real part

XImag[q]-=x[n]\***sin**(2\*pi/N\*q\*n); //imaginary part

}

}

//Printing output

**printf**("\nThe DFT computed is\n");

**for**(W=0;W<n;W++)

**printf**("(%f)+1j\*(%f)\n",XReal[W],XImag[W]);

**for**(E=0;E<n;E++)

{

XMag[E]=**sqrt**(**pow**(XReal[E],2)+**pow**(XImag[E],2)); //magnitude

XPhase[E]=**atan2**(XImag[E],XReal[E]); //phase

}

//Printing phase and Magnitude values

**printf**("The magnitude and phase response of DFT are\n");

**for**(R=0;R<N;R++)

**printf**("XMag[%d]=%f\tXPhase[%d]=%f\n",R,XMag[R],R,XPhase[R]);

}