DSP Final Project Report

MATLAB®

Name: Omar Ahmed Ibrahim

ID : 3954

Name: Mustafa Ibrahim Tohamy

ID : 3851

(1) Sampling and periodicity of sinusoidal signals :

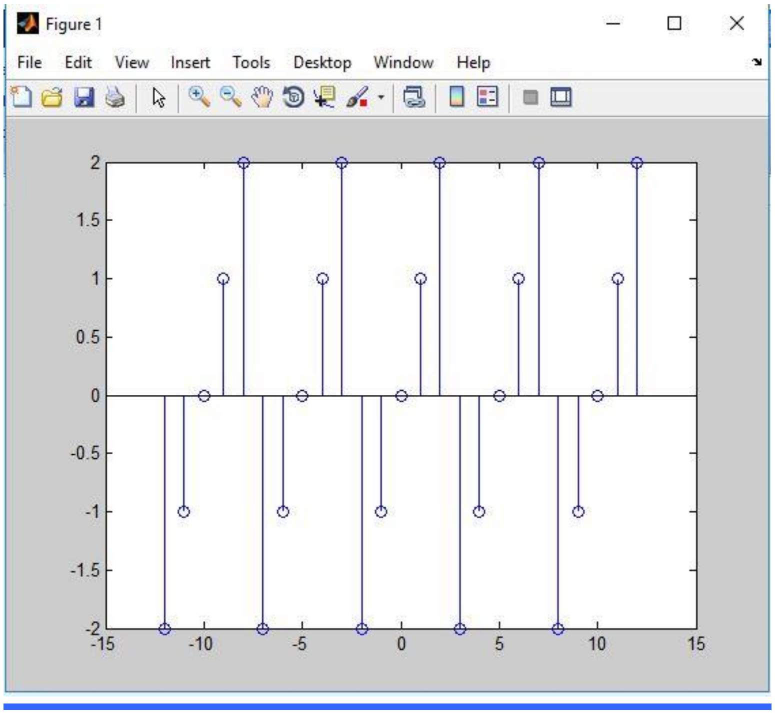
a-

1-

Code:

```
>> n=[-12:1:12];
>>x1n=[-2:1:2,-2:1:2,-2:1:2,-2:1:2,-2:1:2];
>> stem(n,x1n)
```

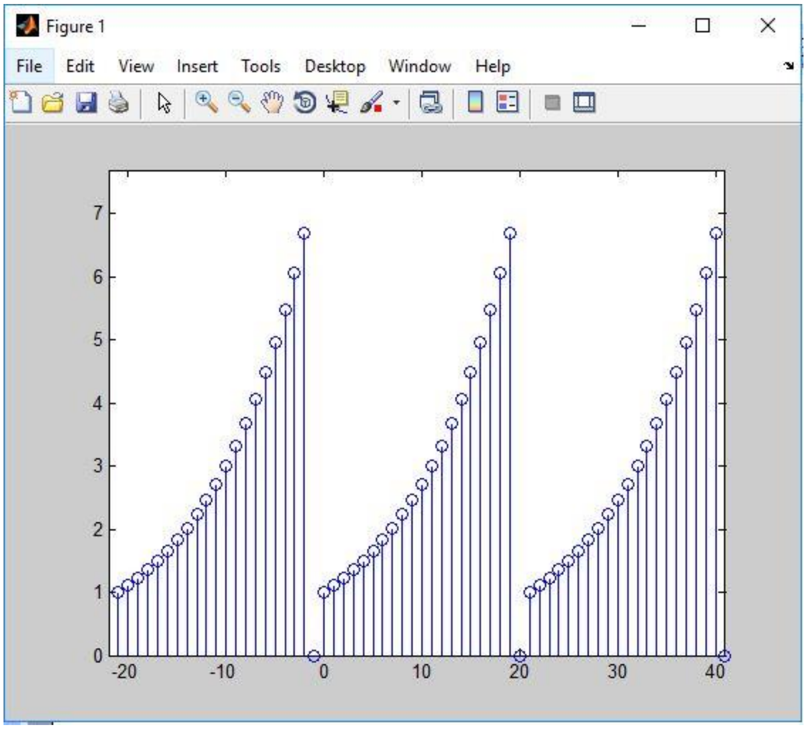
Output:



2-Code:

```
x2n=stepseq(0,0,20)-stepseq(20,0,20);
n=[0:20];
e=exp(0.1*n);
x2n=e.*x2n;
x2n=[x2n,x2n,x2n];
np=[-21:41];
stem(np,x2n);
axis([min(np)-2,max(np)+2,min(x2n),max(x2n)+2]);
```

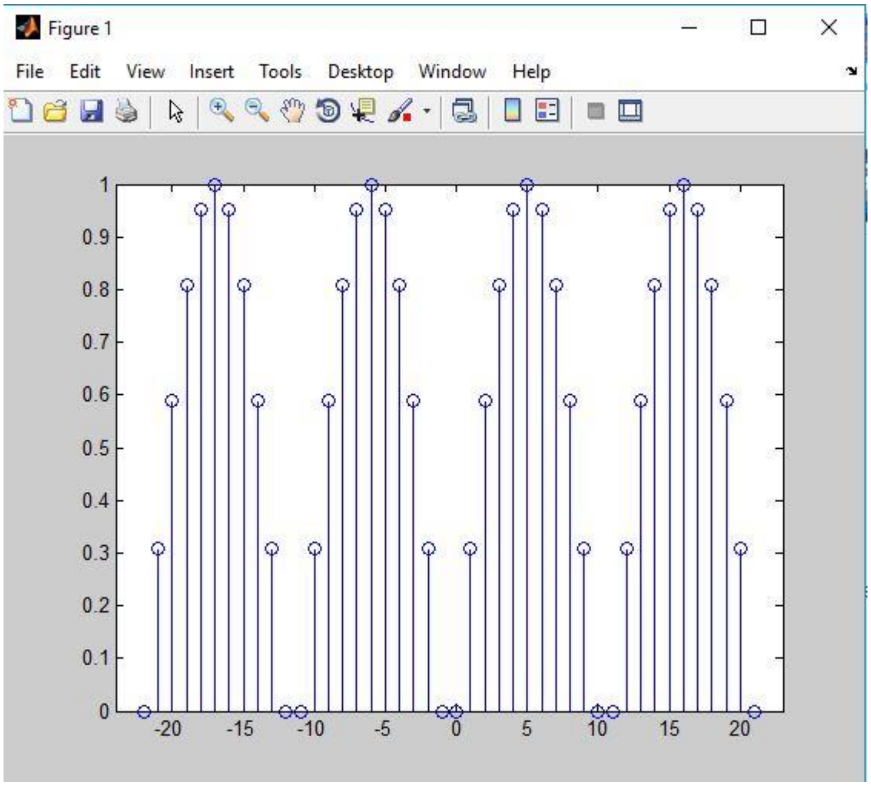
Output:



3-
Code:

```
x3n=stepseq(0,0,10)-stepseq(10,0,10);
n=[0:10];
s=sin(0.1*pi*n);
x3n=s.*(x3n);
x3n=[x3n,x3n,x3n,x3n];
n=[-21:22];
stem(n,x3n);
axis([min(n)-2,max(n)+2,min(x2n),max(x2n)])
```

Output:

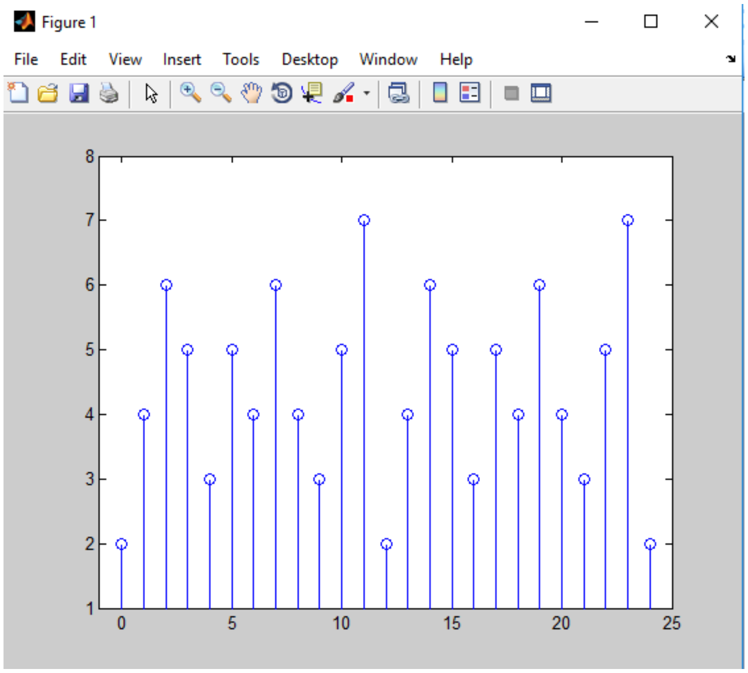


4-
Code:

```
>> n=[0:24];
>>x4a=[1 2 3];
>>x4a=repmat(x4a,[1 9]);
>>x4b=[1 2 3 4];
>>x4b=repmat(x4b,[1 7]);
>>x4=x4a(1:25)+x4b(1:25);
>>stem(n4,x4);
>>axis([min(n4)-1,max(n4)+1,min(x4)-1,max(x4)+1]);
```

Output:

The period of x4(n) is [2 3 4 6 3 5 4 5 4 3 5 7] and there is 2 periods from n = 0:24.



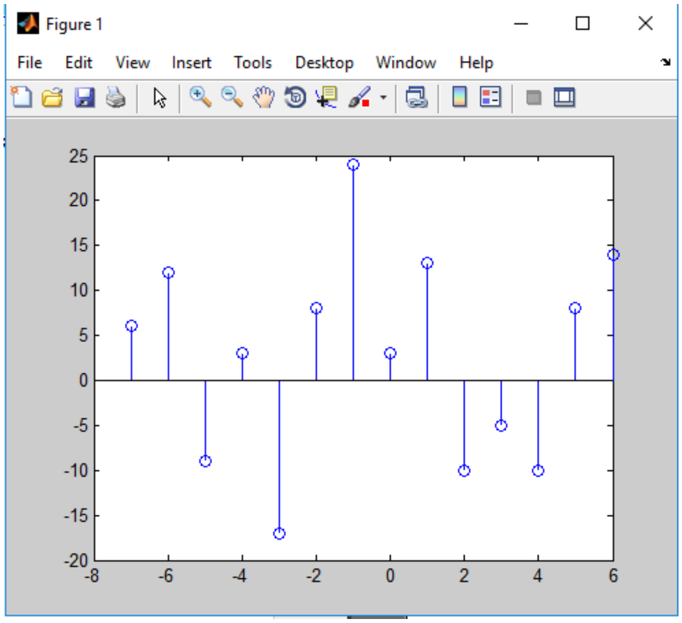
b-

```
>> x=[2 4 -3 1 -5 4 7];
>> n=[-3:3];
```

1) b-Code:

```
>> [x11,n11]=sigshift(x,n,3);
>> [x12,n12]=sigshift(x,n,-4);
>> [x1,n1]=sigadd(2*x11,n11,3*x12,n12);
>> [x1,n1]=sigadd(x1,n1,-x,n);
>> stem(n1,x1);
```

Output:

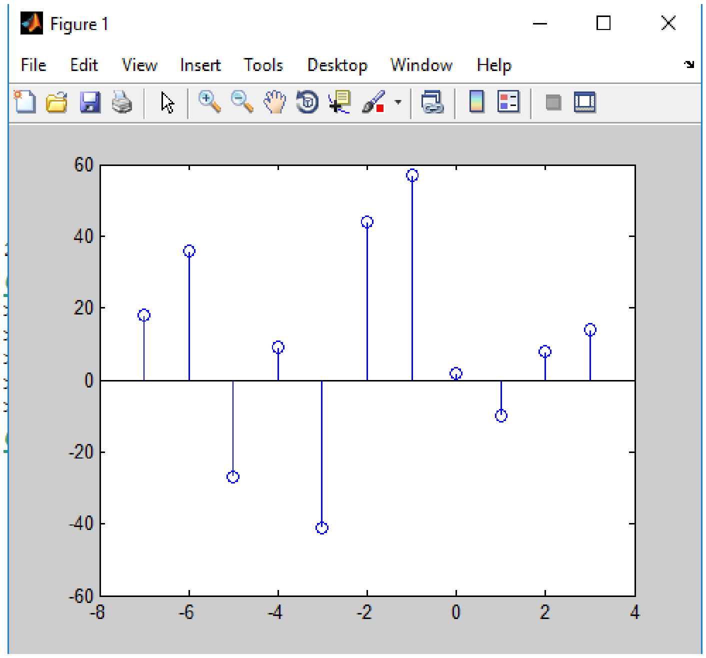


2) b-

Code:

```
>> [x21,n21]=sigshift(x,n,-4);
>> [x22,n22]=sigshift(x,n,-5);
>> [x2,n2]=sigadd(4*x11,n11,5*x12,n12);
>> [x2,n2]=sigadd(x2,n2,2*x,n);
>> stem(n2,x2);
```

Output:

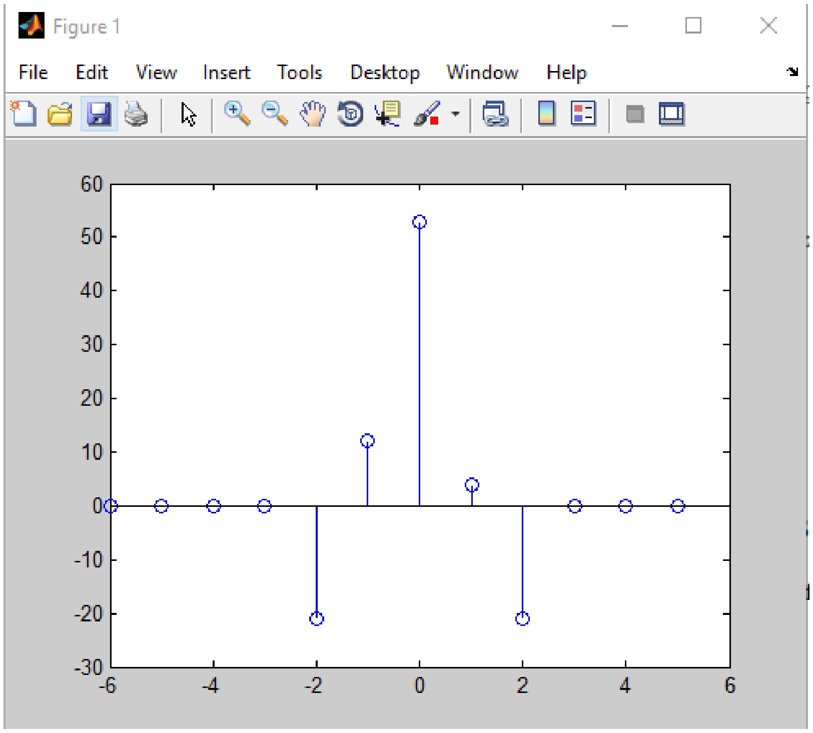


3) b-

Code:

```
>> x=[2 4 -3 1 -5 4 7];
>> n=[-3:3];
>> [x31,n31]=sigshift(x,n,-3);
>> [x32,n32]=sigshift(x,n,2);
>> [x32,n32]=sigmult(x31,n31,x32,n32);
>> [x33,n33]=sigfold(x,n);
>> [x33,n33]=sigshift(x33,n33,1);
>> [x34,n34]=sigshift(x,n,-1);
>> [x33,n33]=sigmult(x33,n33,x34,n34);
>> [x3,n3]=sigadd(x32,n32,x33,n33);
>> stem(n3,x3);
```

Output:

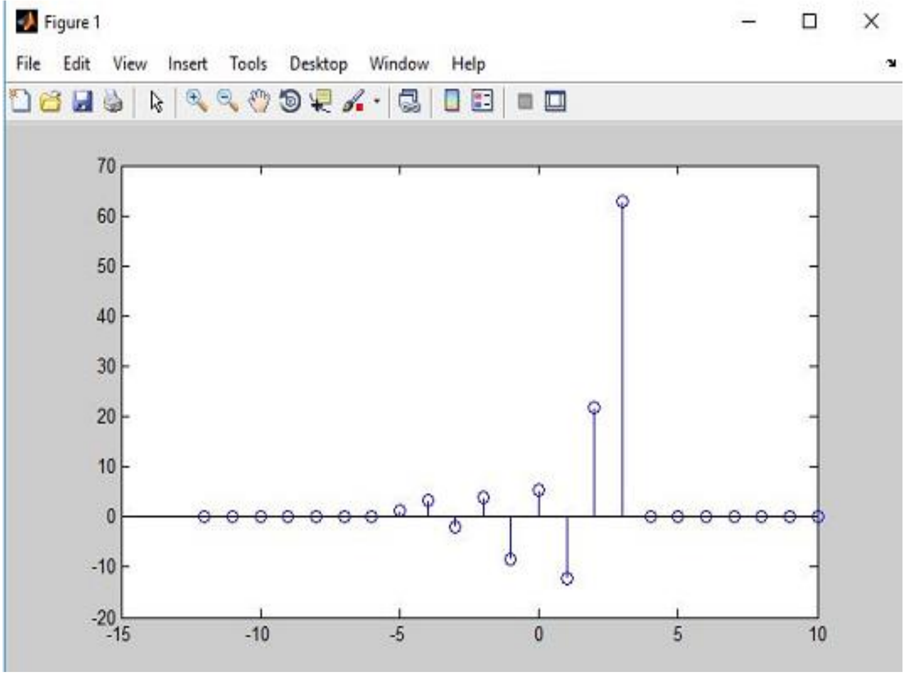


4) b-

Code:

```
>> x=[2 4 -3 1 -5 4 7];
>> n=[-3:3];
>> z=[zeros(1,7)];
>> xz = horzcat(z,x,z);
>> nz=[-10:10];
>> e=2*exp(0.5*nz);
>> c=cos(0.1*pi*nz);
>> [x41,n41]=sigshift(xz,nz,-2);
>> [x4,n4]=sigadd(e.*(xz),nz,c.*(x41),n41);
>> stem(n4,x4);
```

Output:

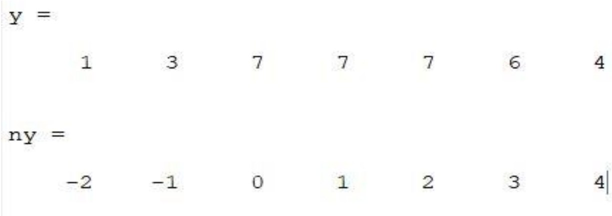


2)Time-Domain Analysis of LTI Systems

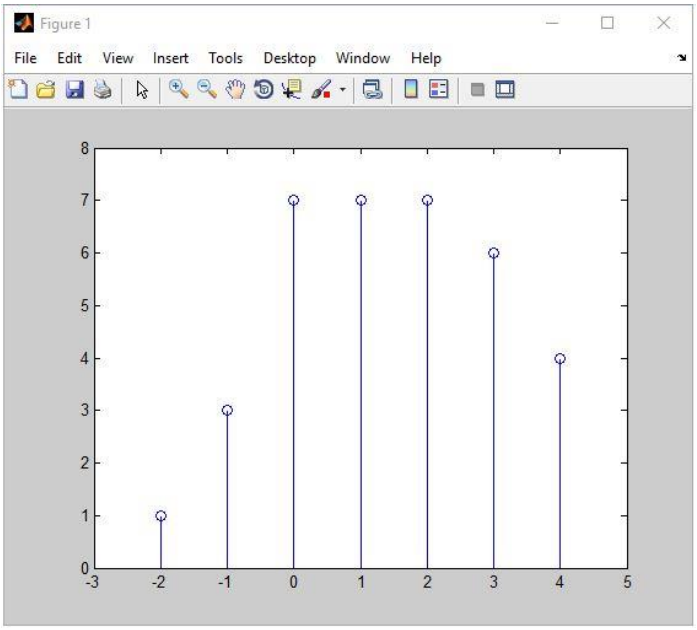
i-Code:

```
x1=[1,2,4];
nx=[0:2];
h1=[ones(1,5)];
nh=[-2:2];
[y,ny]=conv_m(x1,nx,h1,nh)
stem(ny,y)
axis([min(ny)-1,max(ny)+1,min(y)-1,max(y)+1])
```

Output:



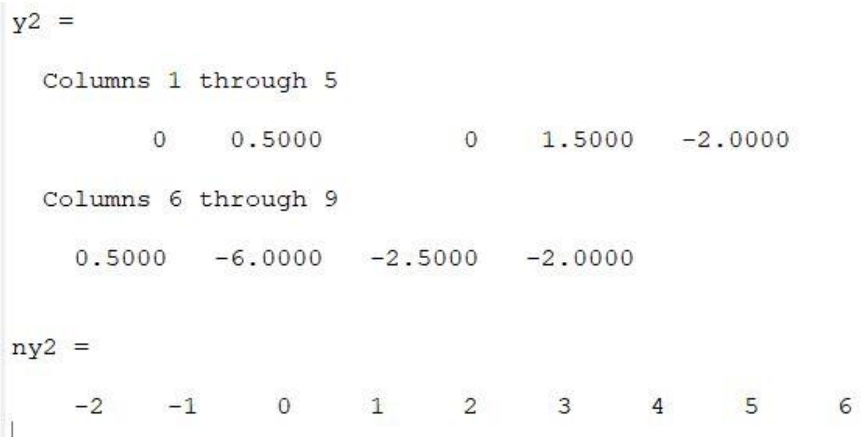
```
y =

     1     3     7     7     7     6     4


ny =

    -2    -1     0     1     2     3     4
```



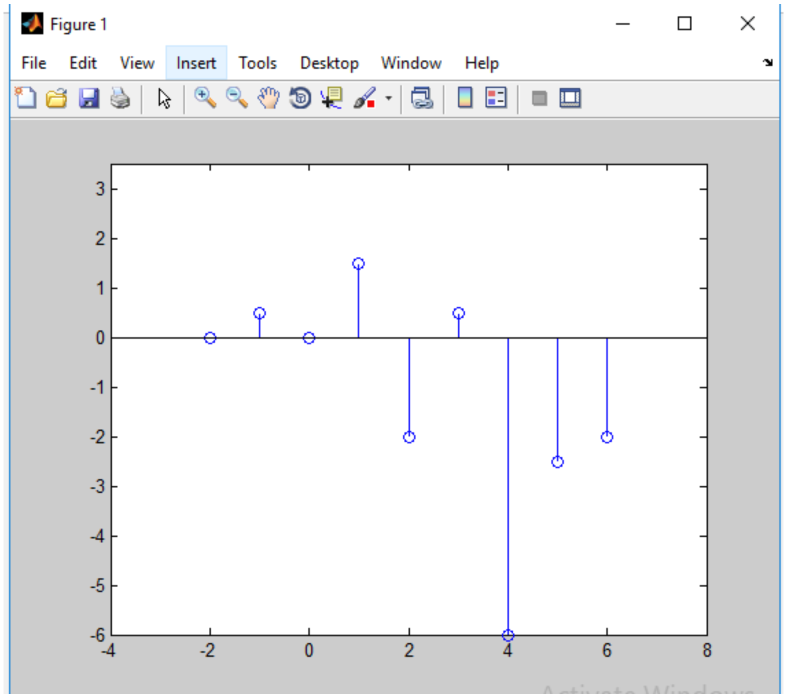
ii-Code:

```
>> x2=[0,1,-2,3,-4];
>> nx2=[0:4];
>> h2=[0.5,1,2,1,0.5];
>> nh2=[-2:2];
>>[y2,ny2]=conv_m(x2,nx2,h2,nh2);
```

Output:



```
y2 =

  Columns 1 through 5

         0    0.5000         0    1.5000   -2.0000

  Columns 6 through 9

    0.5000   -6.0000   -2.5000   -2.0000


ny2 =

    -2    -1     0     1     2     3     4     5     6
```

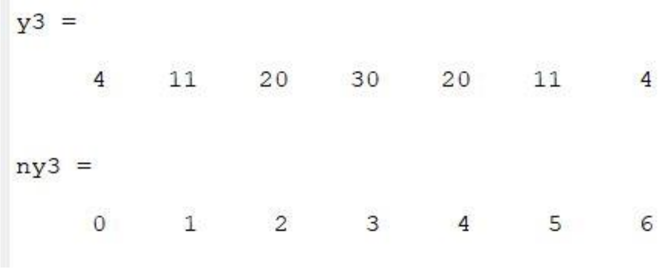


iii-

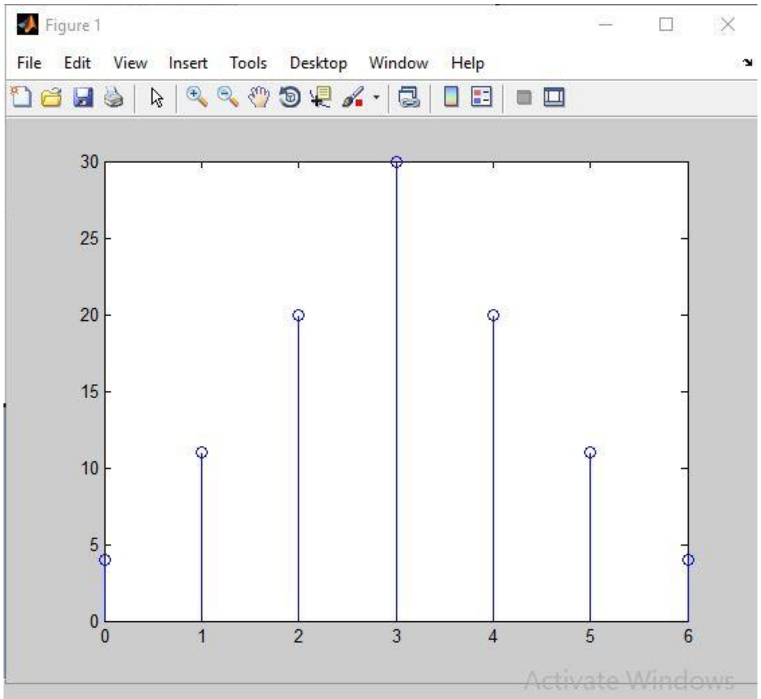
Code:

```
x3=[1,2,3,4];
nx3=[0:3];
h3=[4,3,2,1];
nh3=[0:3];
y3=conv(x3,h3);
ny3=[0:nx3(length(nx3))+nh3(length(nh3))];
stem(ny3,y3);
```

Output:



```
y3 =

     4    11    20    30    20    11     4


ny3 =

     0     1     2     3     4     5     6
```

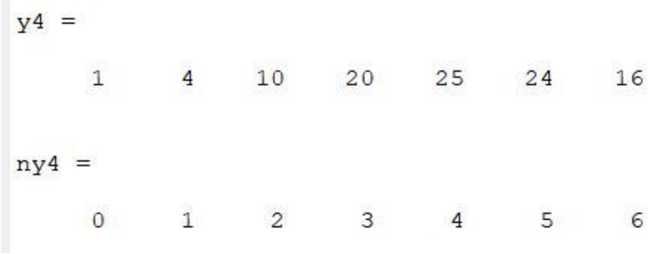


iiii-

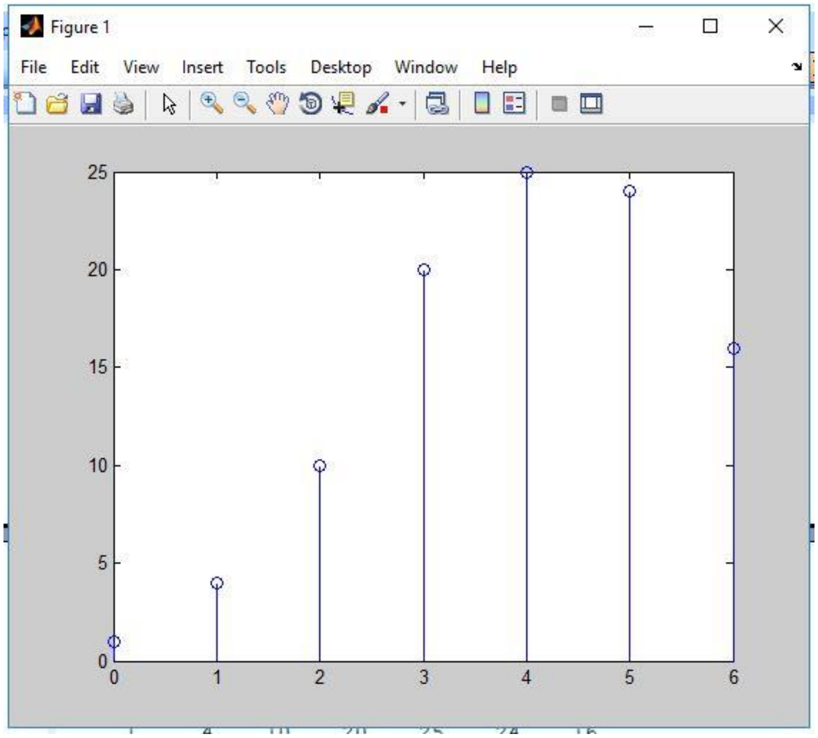
Code:

```
>> x4=[1,2,3,4];
>> nx4=[0:3];
>> h4=[1,2,3,4];
>> nh4=[0:3];
>> y4=conv(x4,h4);
>> stem(ny4,y4);
```

Output:



```
y4 =

     1     4    10    20    25    24    16


ny4 =

     0     1     2     3     4     5     6
```



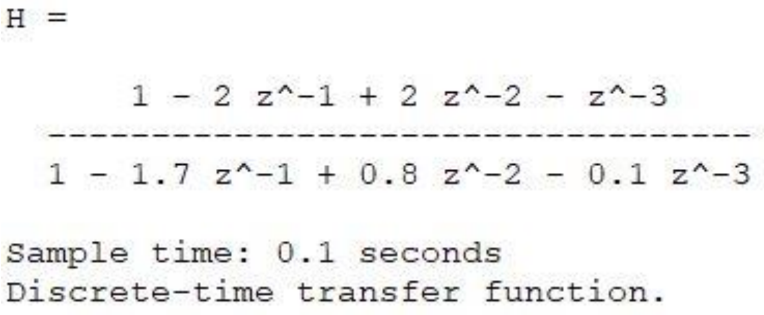
3-Z-Transform analysis of discrete-systems

(a)

i.Code:

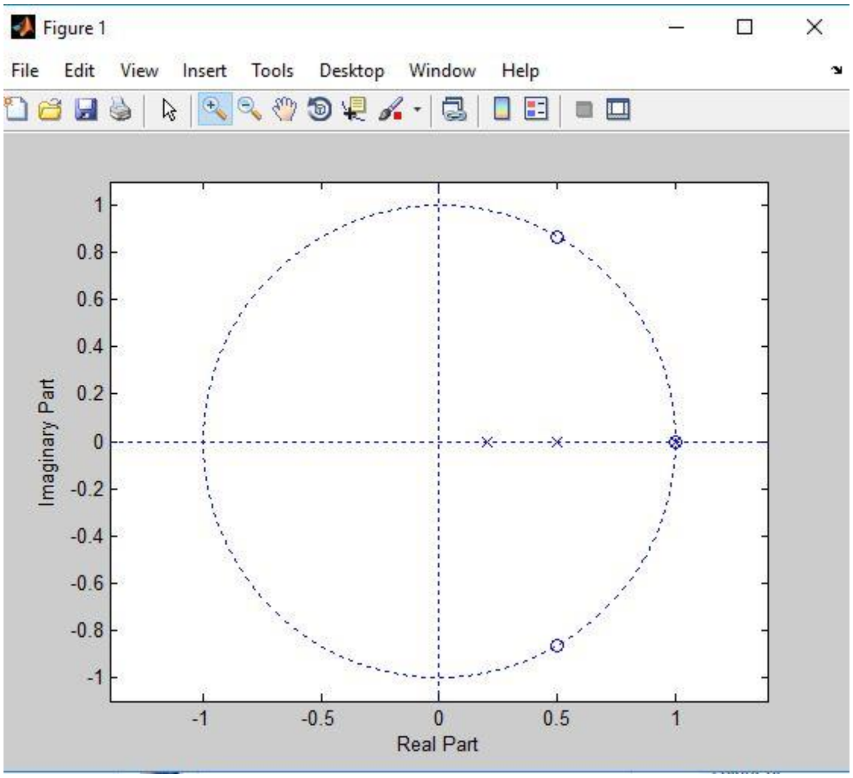
```
>> num=[1 -2 2 -1];
>> den=[1 -1.7 0.8 -0.1];
>> H=tf(num,den,0.1,'Variable','z^-1')
>> zplane(num,den)
```

Output:



```
H =

        1 - 2 z^-1 + 2 z^-2 - z^-3
  ----------------------------------
  1 - 1.7 z^-1 + 0.8 z^-2 - 0.1 z^-3

Sample time: 0.1 seconds
Discrete-time transfer function.
```

1^{st}-Method to check Stability → zplane(num,den)



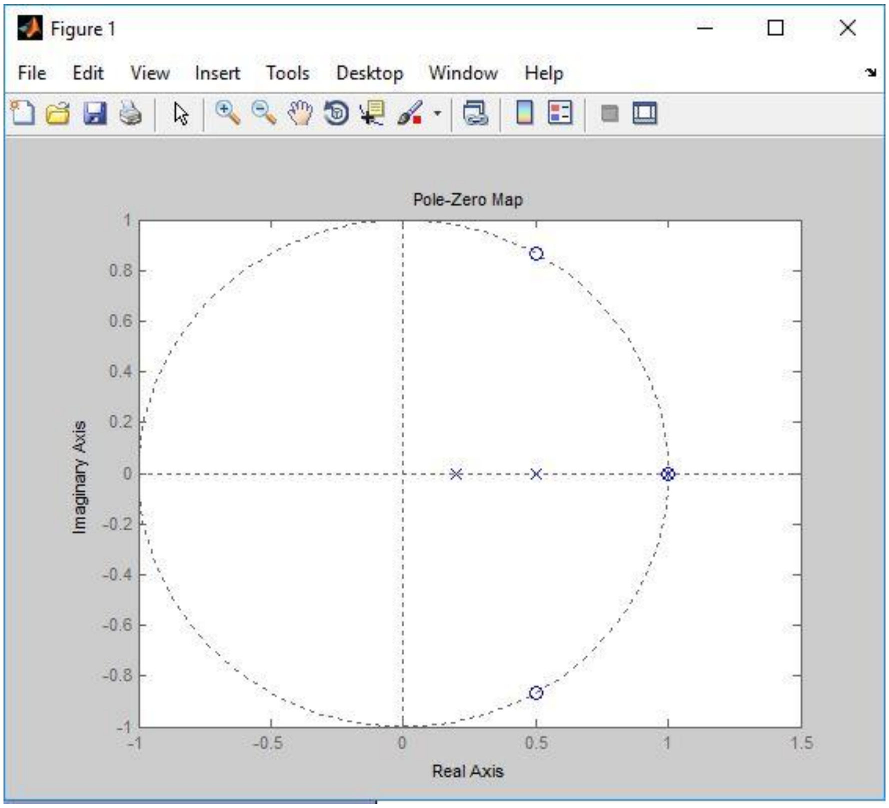
Marginally Stable → as there is one pole on the unit circle | P | =1

2^{nd} Method to check Stability → [p,z]=pzmap(H)

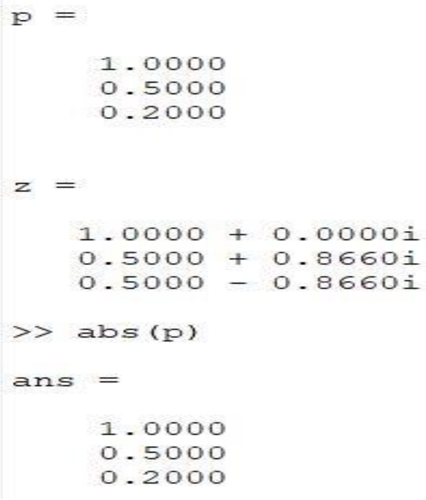
Code :

```
>>[p,z]=pzmap(H)
```

Output:



Marginally Stable→ as there is one pole on the unit circle | P | =1

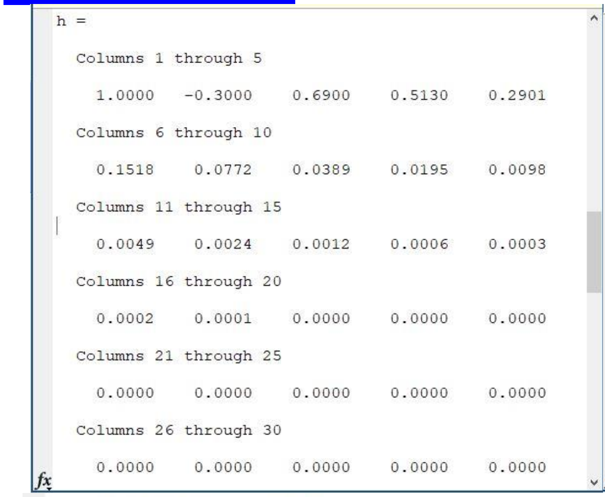


```
p =

    1.0000
    0.5000
    0.2000


z =

   1.0000 + 0.0000i
   0.5000 + 0.8660i
   0.5000 - 0.8660i

>> abs(p)

ans =

    1.0000
    0.5000
    0.2000
```

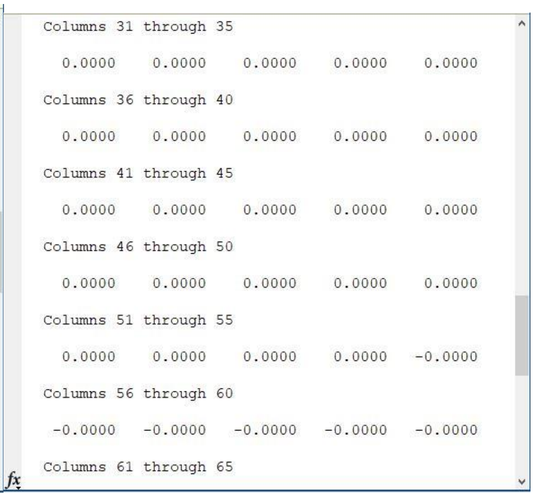
ii-Impulse Response

Code: >> h=filter(num,den,[1 zeros(1,N-1)])

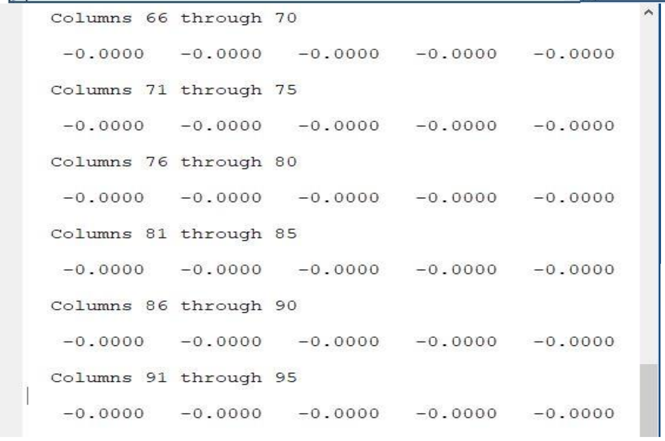
Output:



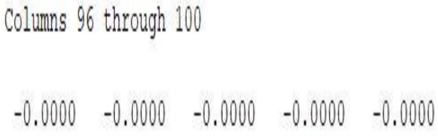
```
h =

  Columns 1 through 5

    1.0000   -0.3000    0.6900    0.5130    0.2901

  Columns 6 through 10

    0.1518    0.0772    0.0389    0.0195    0.0098

  Columns 11 through 15

    0.0049    0.0024    0.0012    0.0006    0.0003

  Columns 16 through 20

    0.0002    0.0001    0.0000    0.0000    0.0000

  Columns 21 through 25

    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 26 through 30

    0.0000    0.0000    0.0000    0.0000    0.0000
```



```
  Columns 31 through 35

    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 36 through 40

    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 41 through 45

    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 46 through 50

    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 51 through 55

    0.0000    0.0000    0.0000    0.0000   -0.0000

  Columns 56 through 60

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 61 through 65
```



```
  Columns 66 through 70

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 71 through 75

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 76 through 80

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 81 through 85

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 86 through 90

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 91 through 95

   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
```



```
Columns 96 through 100

 -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
```

(b).Code:

```
>> num=[0.03 -0.02 0.01];
>> den=[1 -2.8 3.02 -1.468 0.27];
>> H=tf(num,den,0.1,'Variable','z^-1')
```

Output:

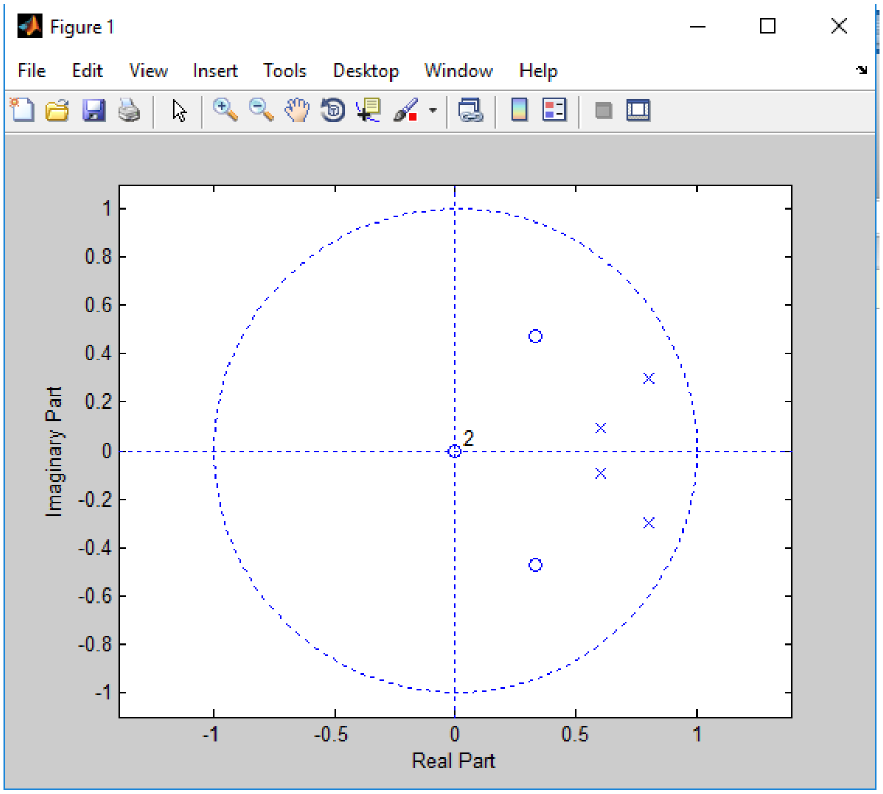
```
H =

             0.03 - 0.02 z^-1 + 0.01 z^-2
  ---------------------------------------------------
  1 - 2.8 z^-1 + 3.02 z^-2 - 1.468 z^-3 + 0.27 z^-4

Sample time: 0.1 seconds
Discrete-time transfer function.
```

Discussing Stability:

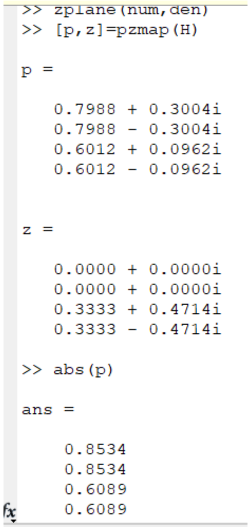
Code:

```
>> zplane(num,den)
[p,z]=pzmap(H)
```

Output:



Stable System since all the poles lies inside the unit circle.|P|<1

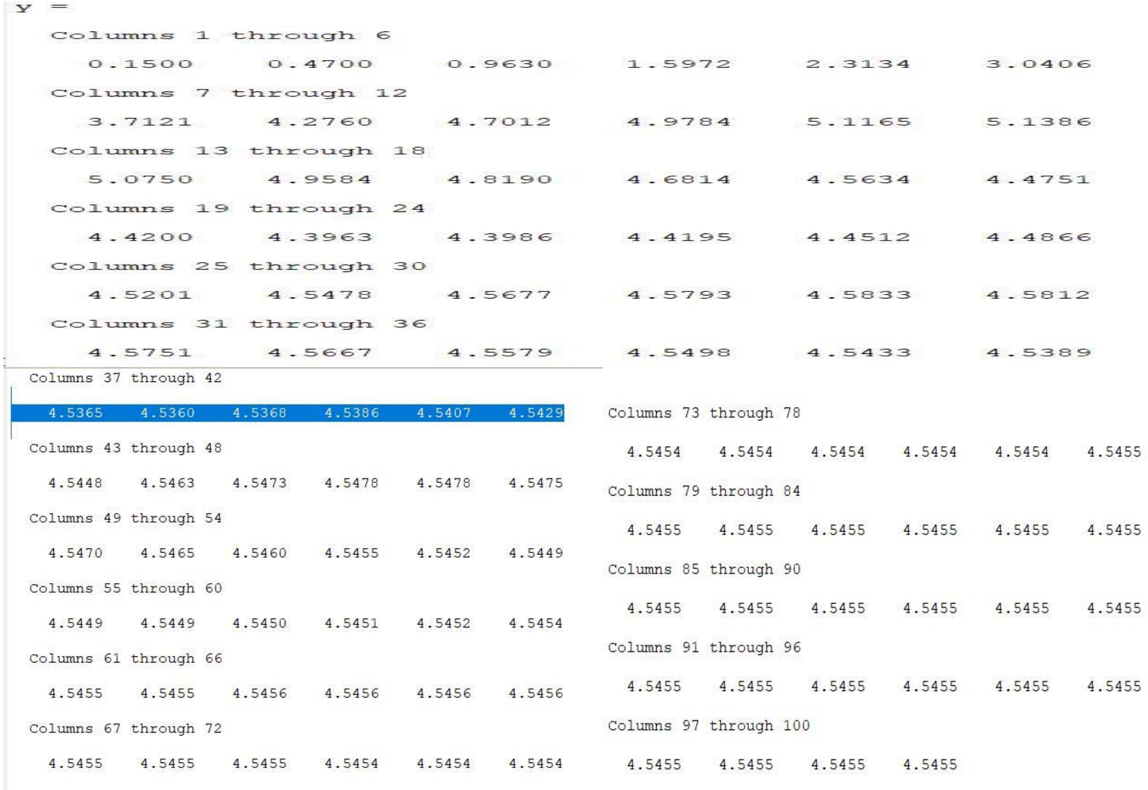


```
>> zplane(num,den)
>> [p,z]=pzmap(H)

p =

   0.7988 + 0.3004i
   0.7988 - 0.3004i
   0.6012 + 0.0962i
   0.6012 - 0.0962i


z =

   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.3333 + 0.4714i
   0.3333 - 0.4714i

>> abs(p)

ans =

    0.8534
    0.8534
    0.6089
fx  0.6089
```

Code : ″ x(n)=5u(n) ″

```
>> N=100;
>> x=5*ones(1,N)
>> y=filter(num,den,x)
```

Output:



```
y =

   Columns 1 through 6

     0.1500      0.4700      0.9630      1.5972      2.3134      3.0406

   Columns 7 through 12

     3.7121      4.2760      4.7012      4.9784      5.1165      5.1386

   Columns 13 through 18

     5.0750      4.9584      4.8190      4.6814      4.5634      4.4751

   Columns 19 through 24

     4.4200      4.3963      4.3986      4.4195      4.4512      4.4866

   Columns 25 through 30

     4.5201      4.5478      4.5677      4.5793      4.5833      4.5812

   Columns 31 through 36

     4.5751      4.5667      4.5579      4.5498      4.5433      4.5389

  Columns 37 through 42

    4.5365    4.5360    4.5368    4.5386    4.5407    4.5429

  Columns 43 through 48

    4.5448    4.5463    4.5473    4.5478    4.5478    4.5475

  Columns 49 through 54

    4.5470    4.5465    4.5460    4.5455    4.5452    4.5449

  Columns 55 through 60

    4.5449    4.5449    4.5450    4.5451    4.5452    4.5454

  Columns 61 through 66

    4.5455    4.5455    4.5456    4.5456    4.5456    4.5456

  Columns 67 through 72

    4.5455    4.5455    4.5455    4.5454    4.5454    4.5454

  Columns 73 through 78

    4.5454    4.5454    4.5454    4.5454    4.5454    4.5455

  Columns 79 through 84

    4.5455    4.5455    4.5455    4.5455    4.5455    4.5455

  Columns 85 through 90

    4.5455    4.5455    4.5455    4.5455    4.5455    4.5455

  Columns 91 through 96

    4.5455    4.5455    4.5455    4.5455    4.5455    4.5455

  Columns 97 through 100

    4.5455    4.5455    4.5455    4.5455
```

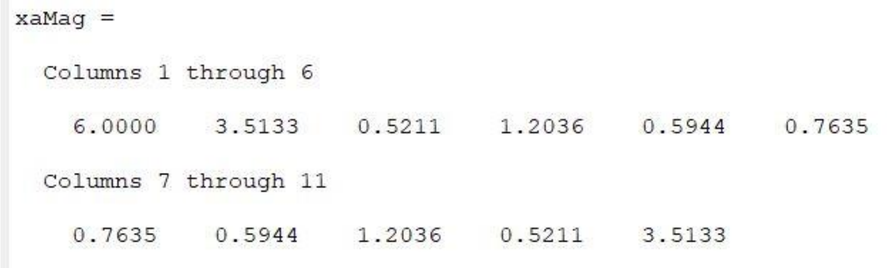
(4) Fourier-transform analysis of discrete systems

a.Code:

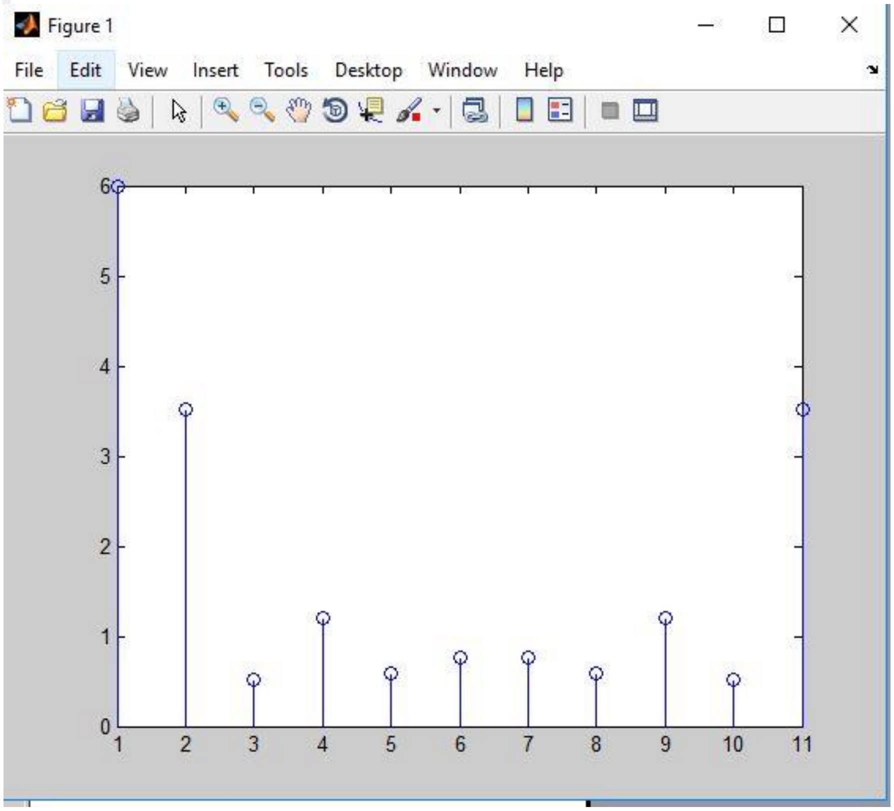
```
na=[0:10];
xa=stepseq(0,0,10)-stepseq(6,0,10);
xa=fft(xa);
xaMag=abs(xa);
stem(xaMag);
xaPhase=angle(xa);
stem(xaPhase);
```

Output:

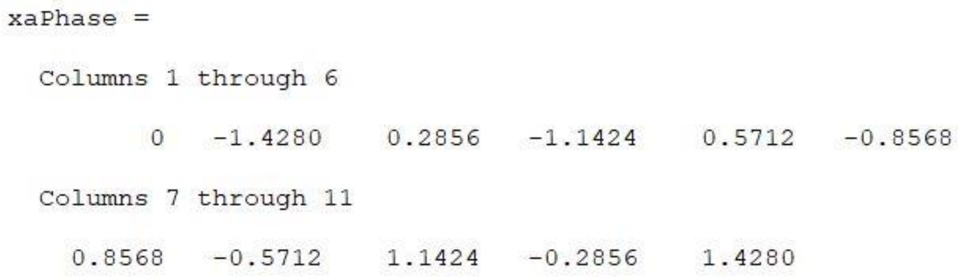
Magnitude:



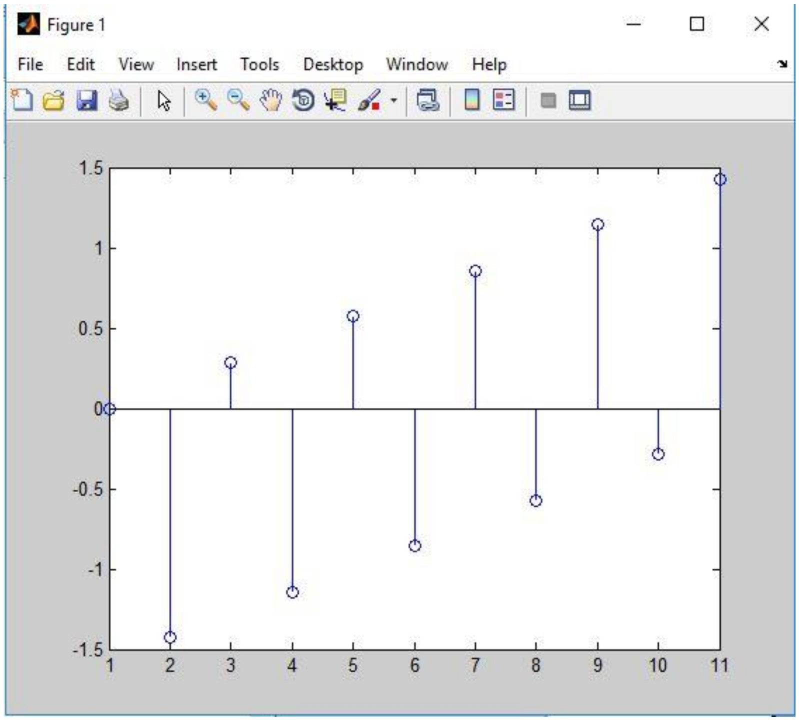
```
xaMag =

  Columns 1 through 6

    6.0000    3.5133    0.5211    1.2036    0.5944    0.7635

  Columns 7 through 11

    0.7635    0.5944    1.2036    0.5211    3.5133
```



Phase:



```
xaPhase =

  Columns 1 through 6

         0   -1.4280    0.2856   -1.1424    0.5712   -0.8568

  Columns 7 through 11

    0.8568   -0.5712    1.1424   -0.2856    1.4280
```

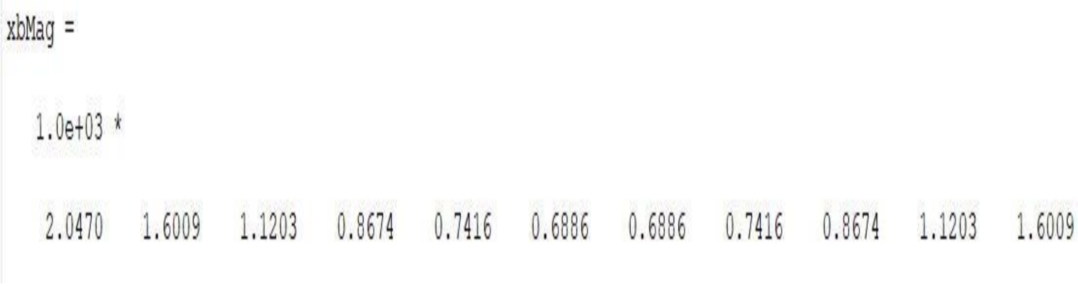


b-Code:

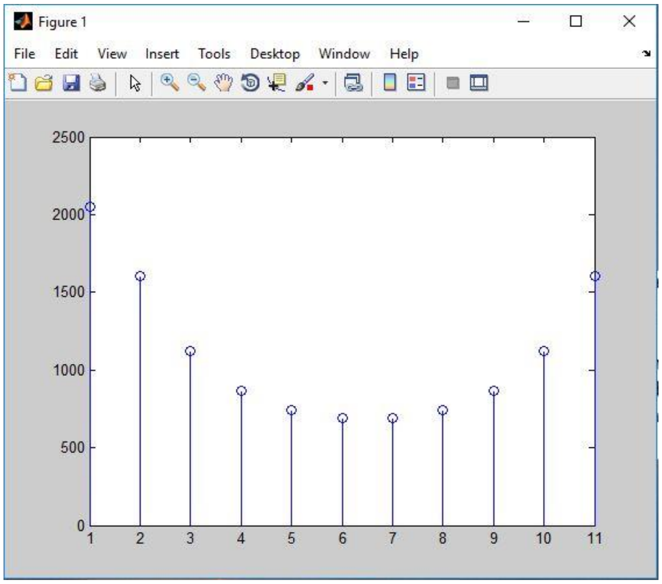
```
>> nb=[0:10];
>> cb=2.^nb;
>> xb=stepseq(0,0,10);
>> [xb,nb]=sigfold(cb.^(xb),nb);
>> xbF=fft(xb);
>> xbMag=abs(xbF);
>> xbPhase=phase(xbF);
```

Output:

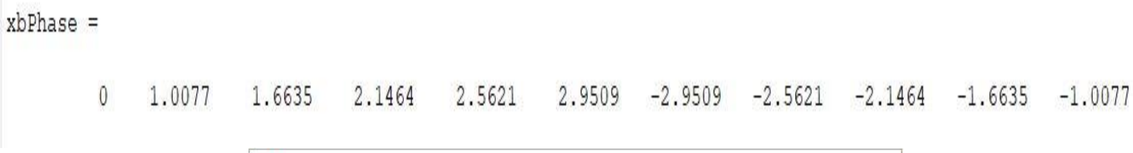
Magnitude:



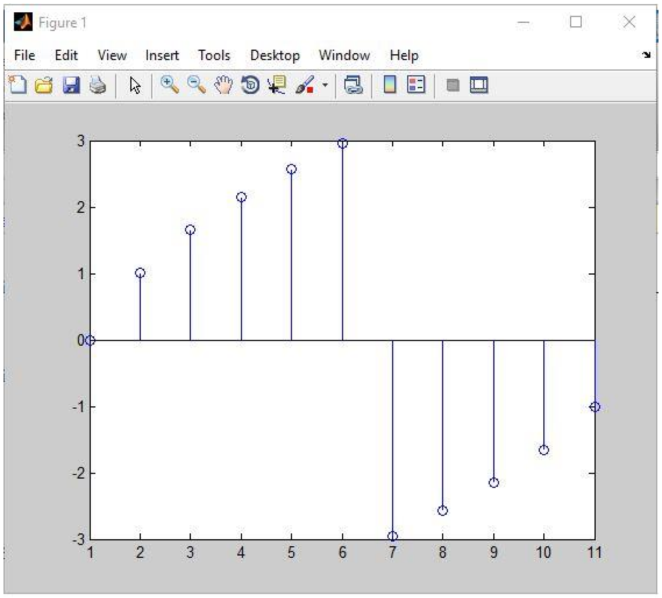
```
xbMag =

   1.0e+03 *

    2.0470    1.6009    1.1203    0.8674    0.7416    0.6886    0.6886    0.7416    0.8674    1.1203    1.6009
```



Phase:



```
xbPhase =

         0    1.0077    1.6635    2.1464    2.5621    2.9509   -2.9509   -2.5621   -2.1464   -1.6635   -1.0077
```

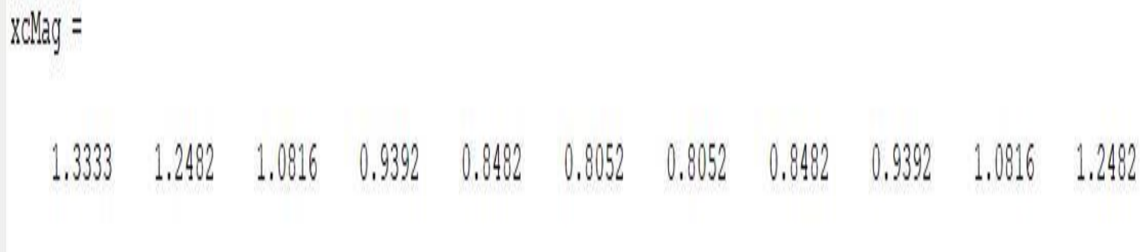


c-Code:

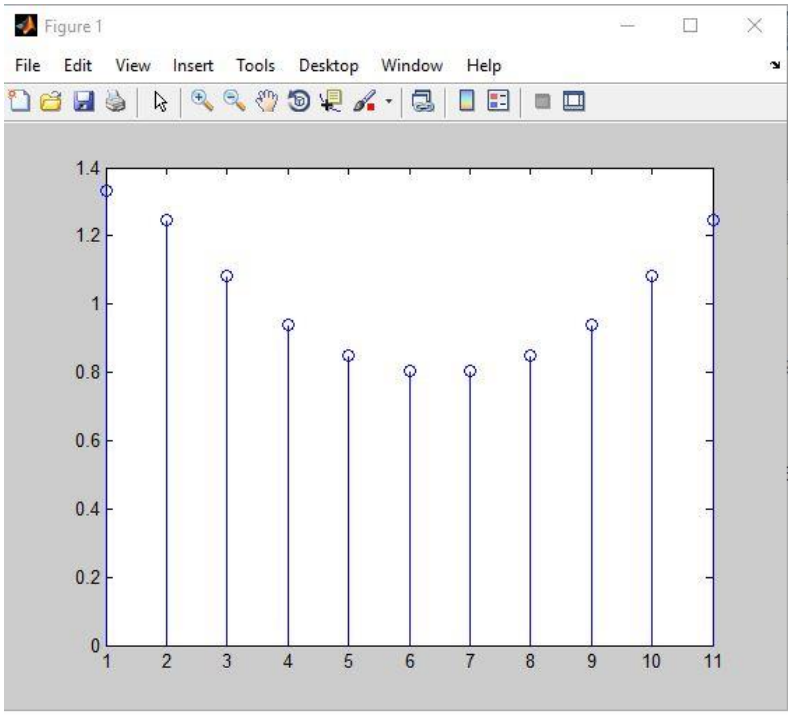
```
>> nc=[0:10];
>> cc=(1/4).^nc;
>> xc=stepseq(0,0,10);
>> xc=cc.^xc;
>> xcF=fft(xc);
>> stem(xcPhase);
>> stem(xcMag);
```

Output:

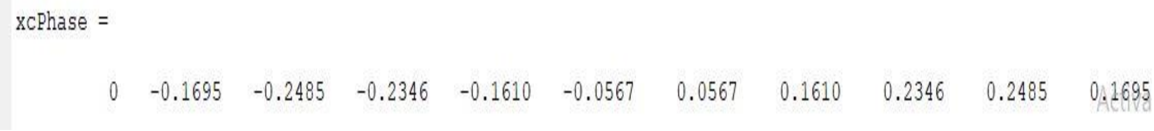
Magnitude:



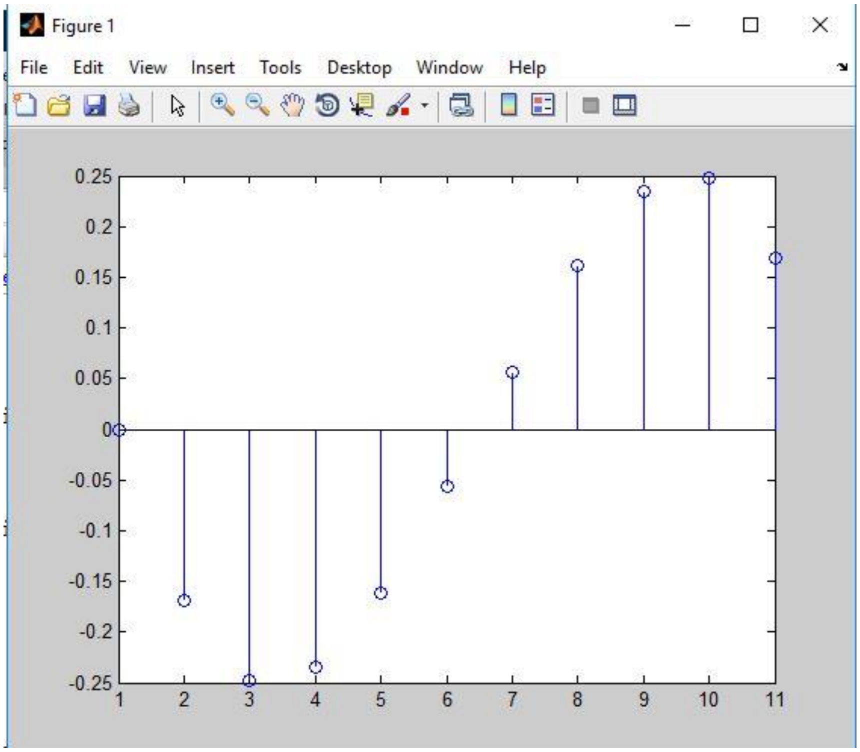
```
xcMag =

   1.3333    1.2482    1.0816    0.9392    0.8482    0.8052    0.8052    0.8482    0.9392    1.0816    1.2482
```



Phase:



```
xcPhase =

        0   -0.1695   -0.2485   -0.2346   -0.1610   -0.0567    0.0567    0.1610    0.2346    0.2485    0.1695
```



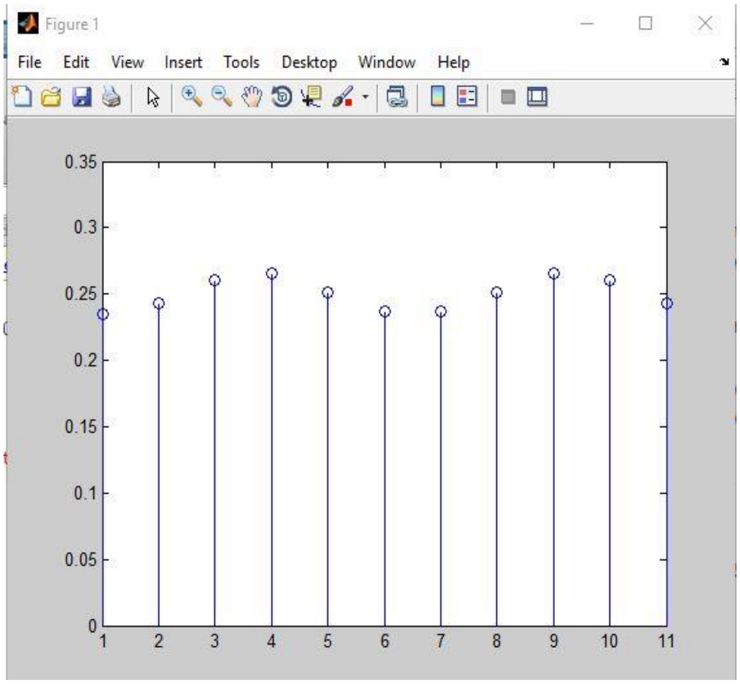
d-Code:

```
nd=[0:10];
cd=  (0.25).^nd;
cs=cd.*sin(2*pi*0.25*nd);
xd=stepseq(0,0,10);
xd=cs.^xd;
xdF=fft(xd);
xdMag=abs(xdF);
xdPhase=phase(xdF);
```

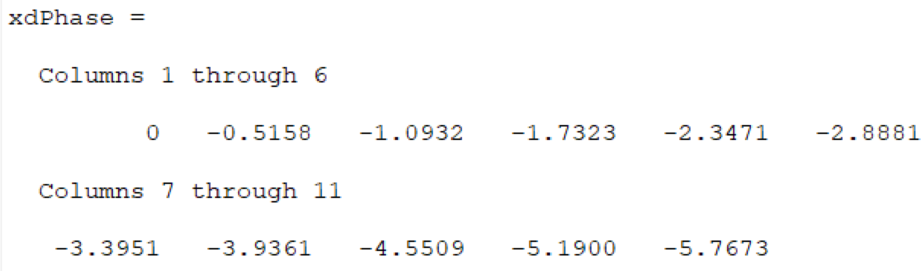
Output:

Magnitude

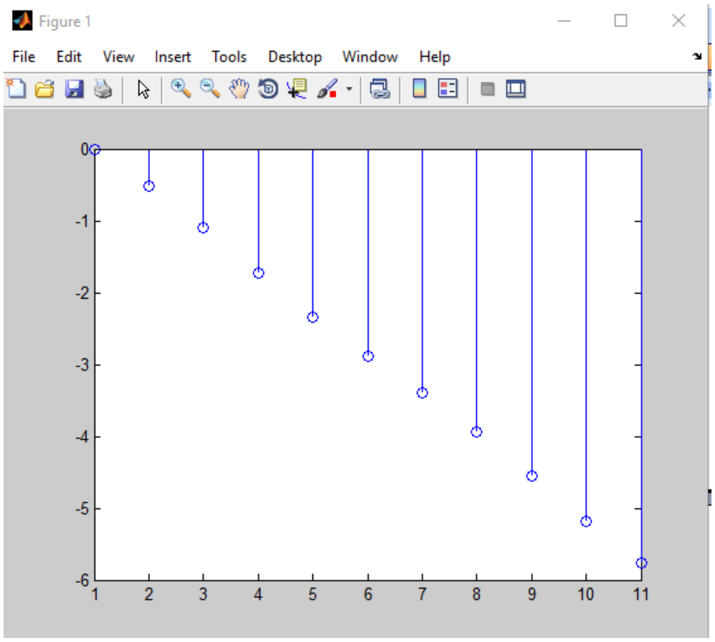
```
  Columns 1 through 6

    0.2353      0.2433      0.2604      0.2659      0.2518      0.2374

  Columns 7 through 11

    0.2374      0.2518      0.2659      0.2604      0.2433
```



Phase:



```
xdPhase =

  Columns 1 through 6

         0   -0.5158   -1.0932   -1.7323   -2.3471   -2.8881

  Columns 7 through 11

   -3.3951   -3.9361   -4.5509   -5.1900   -5.7673
```

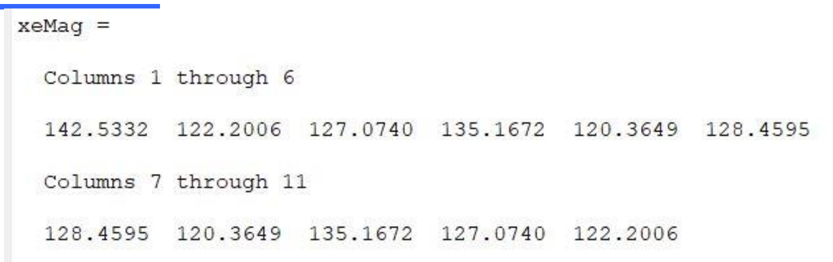


e-Code:

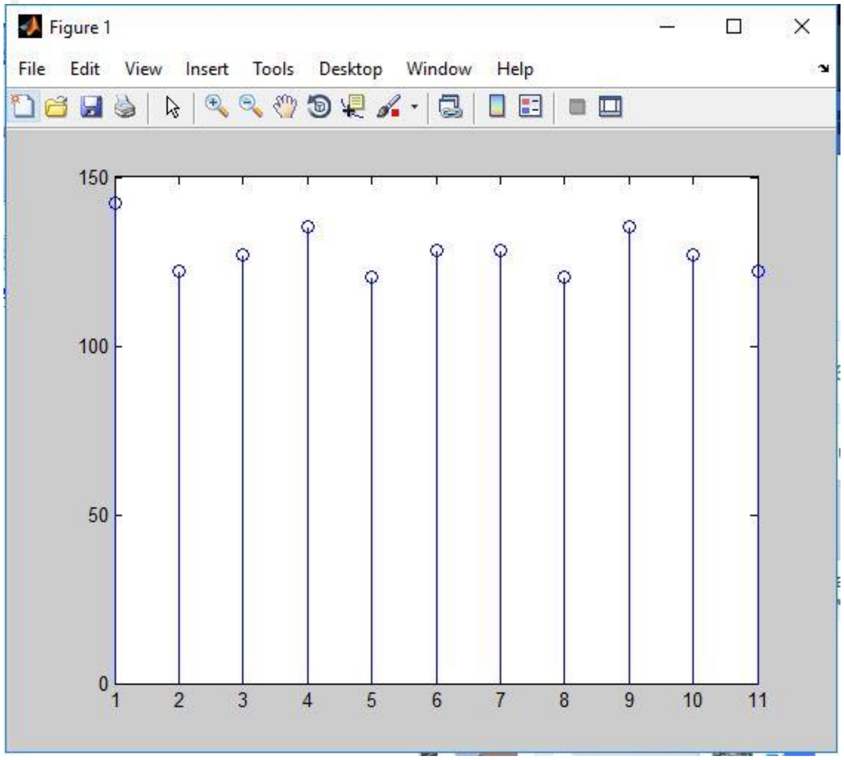
```
>> ne=[0:10];
>> ce=(0.5).^ne;
>> xe=sin(2*pi*0.25*ne) ;
>> xe=ce.^xe;
>> xeF=fft(xe);
>> xeMag=abs(xeF);
>> xePhase=angle(xeF);
>> stem(xeMag);
>> stem(xePhase);
```

Output:

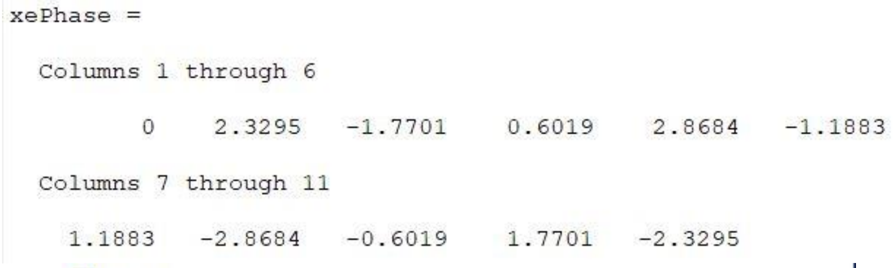
Magnitude:



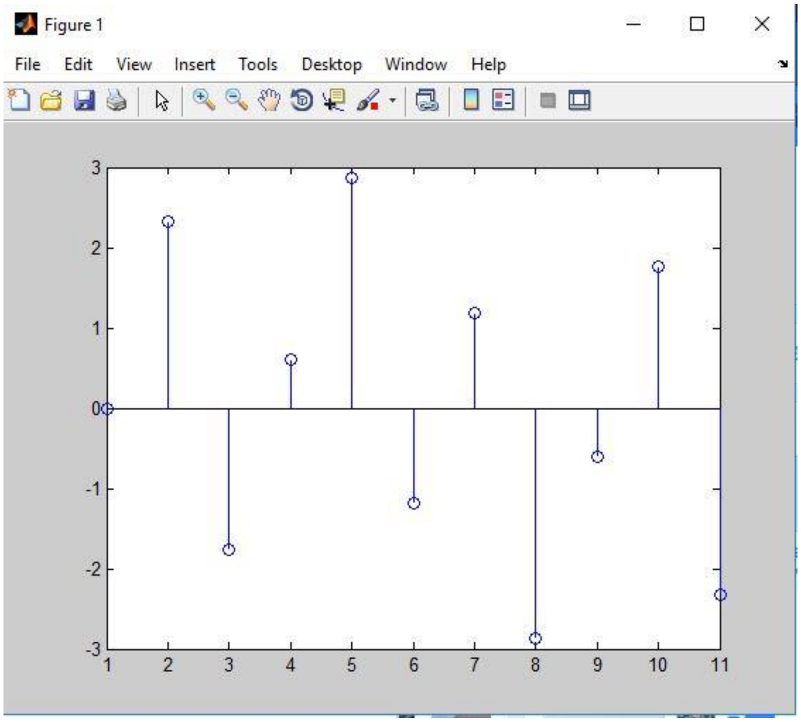
```
xeMag =

  Columns 1 through 6

  142.5332   122.2006   127.0740   135.1672   120.3649   128.4595

  Columns 7 through 11

  128.4595   120.3649   135.1672   127.0740   122.2006
```



Phase:



```
xePhase =

  Columns 1 through 6

         0    2.3295   -1.7701    0.6019    2.8684   -1.1883

  Columns 7 through 11

    1.1883   -2.8684   -0.6019    1.7701   -2.3295
```



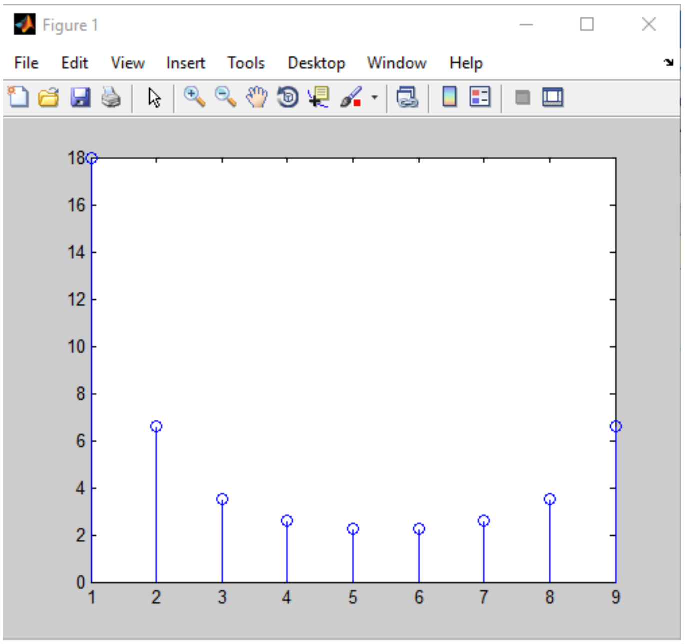
f-Code:

```
>> nf=[-4:4];
>> xf=2-(0.5*nf);
>> xfF=fft(xf);
>> xfMag=abs(xf);
>> xfMag=abs(xfF);
>> xfPhase=phase(xfF);
>> stem(xfMag);
```

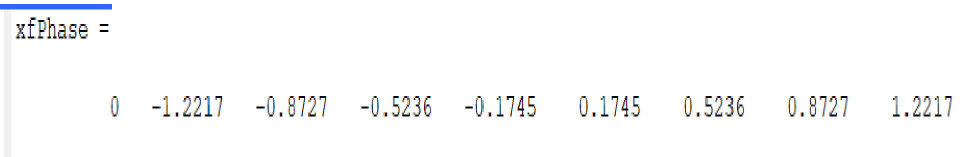
Output:

Magnitude:

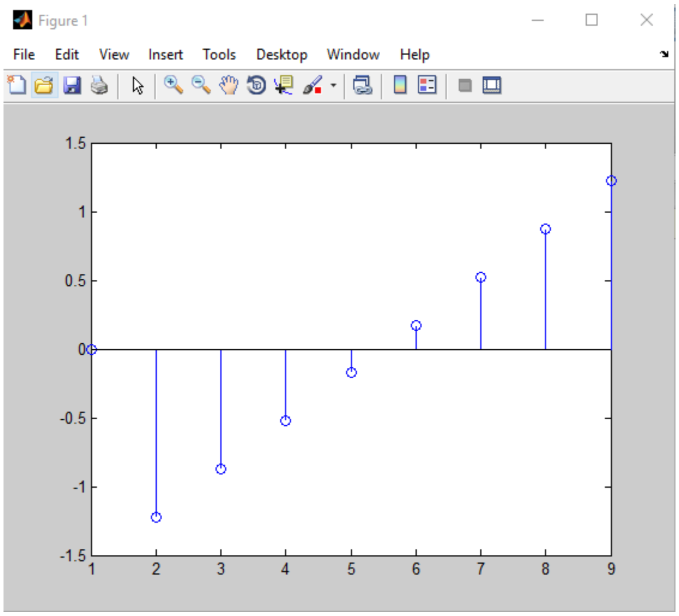
```
xfMag =

   18.0000    6.5786    3.5004    2.5981    2.2847    2.2847    2.5981    3.5004    6.5786
```



Phase:



```
xfPhase =

         0   -1.2217   -0.8727   -0.5236   -0.1745    0.1745    0.5236    0.8727    1.2217
```



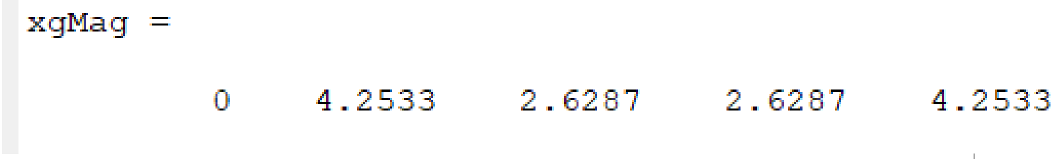
g.

Code:

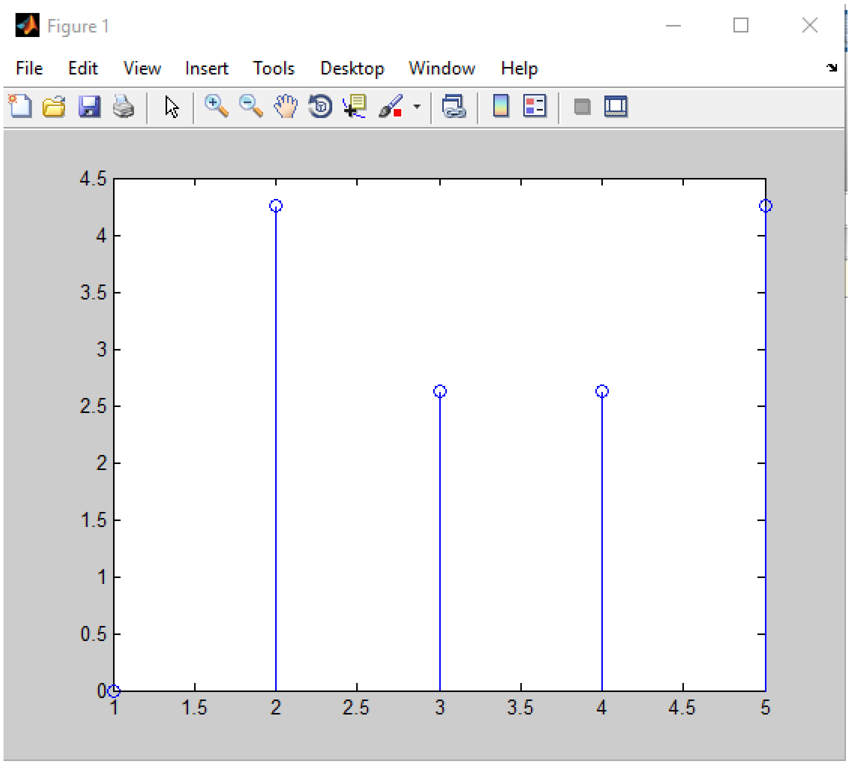
```
>> ng=[-2:2];
>> xg=[-2 -1 0 1 2];
>> xgF=fft(xg);
>> xgMag=abs(xgF);
>> xgPhase=phase(xgF);
>> stem(xgMag);
>> stem(xgPhase);
```

Output:

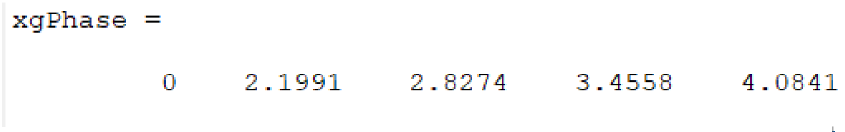
Magnitude:



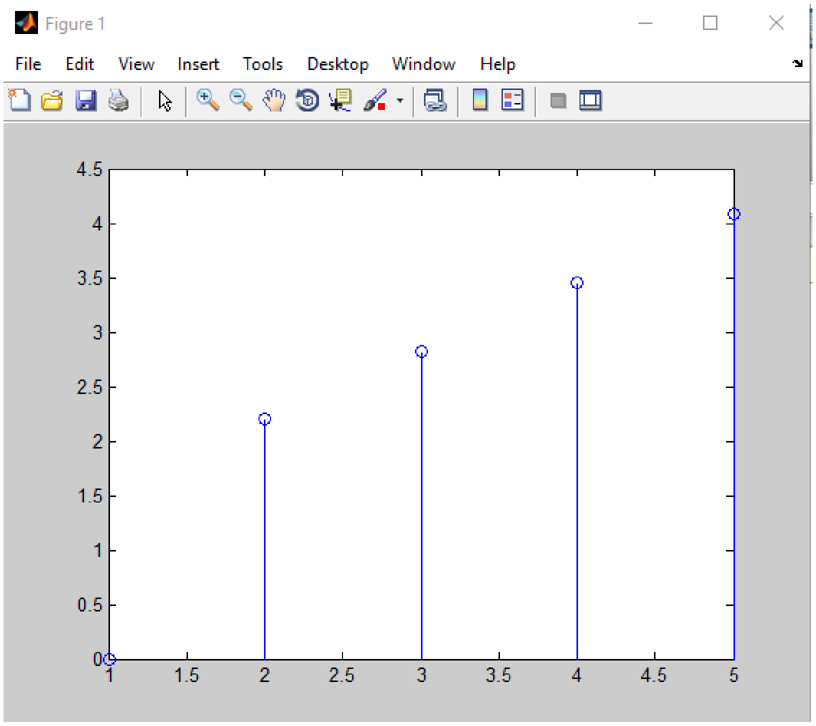
```
xgMag =

         0    4.2533    2.6287    2.6287    4.2533
```



Phase:



```
xgPhase =

         0    2.1991    2.8274    3.4558    4.0841
```



End of Report

DSP Final MATLAB Project