GENERATION OF ELEMENTARY DISCRETE - TIME SEQUENCES

PROGRAM:

```
clc;
clear all;
close all;

%UNIT IMPULSE SIGNAL%
N=8;
n=0;
x=ones(1,1);
subplot(3,3,1);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('UNIT IMPULSE SIGNAL');

%UNIT STEP SIGNAL%
N=8;
n=0:1:N-1;
x=ones(1,N);
subplot(3,3,2);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('UNIT STEP SIGNAL');

%UNIT RAMP SIGNAL%
N=8;
n=0:1:N-1;
x=0:1:N-1;
subplot(3,3,3);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('UNIT RAMP SIGNAL');

%EXPONENTIAL WAVE%
N=8;
n=0:1:N-1;
x=exp(n);
subplot(3,3,4);
stem(n,x );
xlabel('n');
ylabel('x(n)');
title('EXPONENTIAL WAVEFORM');
```

```
%SINE WAVE%
N=8;
n=0:1:N-1;
x=sin(.2*pi*n);
subplot(3,3,5);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('SINE WAVE');

%COSINE WAVE%
N=8;
n=0:1:N-1;
x=cos(.2*pi*n);
subplot(3,3,6);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('COSINE WAVE');
```

PROGRAM:

```
clc;
clear all;
close all;
x=input('Enter the first input sequence x(n):');
h=input('Enter the second input sequence h(n):');
n1=length(x);
n2=length(h);
n=n1+n2-1;
X=fft(x,n);
H=fft(h,n);
Y=X.*H;
y=ifft(Y);
disp('linear convolution output is:');
disp(y);
t1=0:n1-1;
subplot(1,3,1);
stem(t1,x);
xlabel('n');
ylabel('Amplitude');
title('first input sequence');
t2=0:n2-1;
subplot(1,3,2);
stem(t2,h);
xlabel('n');
ylabel('Amplitude');
title('second input sequence');
t=0:1:n-1;
subplot(1,3,3);
stem(t,y);
xlabel('n');
ylabel('Amplitude');
title('output sequence');
```

PROGRAM

```
clc;
clear all;
close all;
x1=input('enter the 1st input sequence');
x2=input('enter the 2nd input sequence');
n1=length(x1);
n2=length(x2);
if(n1<n2)
   x1=[zeros(1,n2-n1)];
elseif(n2<n1)
   x2=[zeros(1,n1-n2)];
else
   x1=x1;
   x2=x2;
end;
n1=length(x1);
n2=length(x2);
A=fft(x1,n1);
B=fft(x2,n2);
Y=A.*B;
y=ifft(Y);
n=length(y);
disp('circular convolution output is:');
disp(y);
t1=0:n1-1;
subplot(1,3,1);
stem(t1,x1);
xlabel('n-->');
ylabel('amplitude-->');
title('first input sequence');
t2=0:n2-1;
subplot(1,3,2);
stem(t2,x2);
xlabel('n-->');
ylabel('amplitude-->');
title('second input sequence');
t=0:1:n-1;
subplot(1,3,3);
stem(t,y);
xlabel('n-->');
ylabel('amplitude-->');
title('output sequence');
```

PROGRAM:

```
%AUTOCORRELATION%
clc;
clear all;
close all;
x=input('Enter the sequence');
y=xcorr(x,x);
figure;
subplot(2,1,1);
stem(x);
ylabel('amplitude');
xlabel('n');
subplot(2,1,2);
stem(fliplr(y));
y
xlabel('n');
ylabel('amplitude');
disp('the resultant signal is  ');
fliplr(y);
```

```
%CROSS CORRELATION%
clc;
close all;
clear all;
x=input('Enter the first sequence');
h=input('Enter the second sequence');
y=xcorr(x,h);
figure;
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('amplitude');
subplot(3,1,2);
stem(h);
ylabel('amplitude');
xlabel('n');
subplot(3,1,3);
stem(fliplr(y));
y
ylabel('amplitude');
xlabel('n');
title('The resultant signal is ');
fliplr(y)
```

PROGRAM:

```
clc;
clear all;
x=input ('Enter the sequence :');
n=length (x);
y=fft(x,n);
di sp('output sequence is:');
disp(y);
m=abs(y);
disp('magnitude function is:');
disp(m);
p=angle(y);
disp('phase function is:');
disp(p);
subplot(1,3,1);
t1=0 : n-1;
stem(t1,x);
xlabel('n-->');
ylabel('amplitude -->');
title('input sequence');
subplot(1,3,2);
t2=0 : n-1;
stem (t2,m);
xlabel('n-->');
ylabel('amplitude -->');
title('magnitude plot');
subplot(1,3,3);
t3=0 : n-1;
stem(t3,p);
xlabel('n-->');
ylabel('phase -->');
title('phase plot');
```

PROGRAM:

```
clc;
clear all;
close all;
N=input('enter the order of filter');
wc=(pi/2);
a=(N-1)/2;
for n=1:N
   if((n-1)==a)
      hd(n)=(wc/pi);
   else
      hd(n)=(sin(wc*(n-1-a)))/(pi*(n-1-a));
   end;
   w(n)=1;
end;
h=w.*hd;
a=0:0.01:pi;
b=freqz(h,1,a);
mag=20*log(abs(b));
plot(a/pi,mag);
grid;
xlabel('normalized frequency\omega^pi');
ylabel('magnitude in db');
title('low pass filter');
```

PROGRAM:

```
clc;
clear all;
close all;
N=input('enter the order of filter');
wc=(pi/2);
a=(N-1)/2;
for n=1:N
   if((n-1)==a)
      hd(n)=(pi-wc)/pi;
   else
      hd(n)=sin(pi*(n-1-a))-sin(wc*(n-1-a))/(pi*(n-1-a));
   end;
   w(n)=0.54-(0.46*cos(2*pi*(n-1)/(N-1)));
end;
h=w.*hd;
a=0:0.01:pi;
b=freqz(h,1,a);
mag=20*log(abs(b));
plot(a/pi,mag);
grid;
xlabel('normalized frequency\omega^pi');
ylabel('magnitude in db');
title('high pass filter');
```

PROGRAM:

```
clc;
clear all;
close all;
N=input('enter the order of filter');
wc1=(pi/4);
wc2=(3*pi)/4;
a=(N-1)/2;
for n=1:N
   if((n-1)==a)
      hd(n)=(wc2-wc1)/pi;
   else
     hd(n)=(sin(wc2*(n-1-a))-sin(wc1*(n-1-a)))/(pi*(n-1-a));
   end;
   w(n)=0.5+(0.5*cos(2*pi*(n-1-a)/(N-1)));
end;
h=w.*hd;
a=0:0.01:pi;
b=freqz(h,1,a);
mag=20*log(abs(b));
plot(a/pi,mag);
grid;
xlabel('normalized frequency\omega^pi');
ylabel('magnitude in db');
title('bandpass filter');
```

PROGRAM:

```
clc;
clear all;
close all;
N=input('enter the order of filter');
wc1=(pi/4);
wc2=(3*pi)/4;
a=(N-1)/2;
for n=1:N
   if((n-1)==a)
      hd(n)=1-(wc1+pi-wc2)/pi;
   else
     hd(n)=(sin(wc1*(n-1-a))-sin(wc2*(n-1-a)))/(pi*(n-1-a));
   end;
   w(n)=0.5+(0.5*cos(2*pi*(n-1-a)/(N-1)));
end;
h=w.*hd;
a=0:0.01:pi;
b=freqz(h,1,a);
mag=20*log(abs(b));
plot(a/pi,mag);
grid;
xlabel('normalized frequency\omega^pi');
ylabel('magnitude in db');
title('bandstop filter');
```

PROGRAM:

```
clc;
clear all;
close all;
ap=0.5;
as=50;
fp=1000;
fs=2000;
f=5000;
wp=2*fp/f;
ws=2*fs/f;
wn=[wp,ws];

%TO FIND THE CUT OFF FREQ & ORDER OF FILTER

[n,wn]=buttord(wp,ws,ap,as);
[b,a]=butter(n,wn,'low');
w=0:0.01:pi;
h=freqz(b,a,w);
p=angle(h);
mag=20*log10(abs(h));
subplot(2,1,1);
plot(w/pi,mag);
grid;
xlabel('normalized frequency');
ylabel('magnitude in db-->');
title('magnitude response(low pass filter)');
subplot(2,1,2);
plot(w/pi,p);
grid;
xlabel('normalized frequency-->');
ylabel('phase in radians-->');
title('phase response(low pass filter)');
```

PROGRAM:

```
clc;
clear all;
close all;
ap=0.5;
as=50;
fp=1000;
fs=2000;
f=5000;
wp=2*fp/f;
ws=2*fs/f;
wn=[wp,ws];

%TO FIND THE CUT OFF FREQ&ORDER OF FILTER
[n,wn]=buttord(wp,ws,ap,as);
[b,a]=butter(n,wn,'high');
w=0:0.01:pi;
h=freqz(b,a,w);
p=angle(h);
mag=20*log10(abs(h));
subplot(2,1,1);
plot(w/pi,mag);
grid;
xlabel('normalized frequency');
ylabel('magnitude in db-->');
title('magnitude response(highpass filter)');
subplot(2,1,2);
plot(w/pi,p);
grid;
xlabel('normalized frequency-->');
ylabel('phase in radians-->');
title('phase response(highpass filter)');
```

PROGRAM:

```
clc;
clear all;
close all;
ap=0.5;
as=50;
fp=1000;
fs=2000;
f=5000;
wp=2*fp/f;
ws=2*fs/f;
wn=[wp,ws];
[n]=buttord(wp,ws,ap,as);
[b,a]=butter(n,wn,'bandpass');
w=0:0.01:pi;
h=freqz(b,a,w);
p=angle(h);
mag=20*log10(abs(h));
subplot(2,1,1);
plot(w/pi,mag);
grid;
xlabel('normalized frequency');
ylabel('magnitude in db-->');
title('magnitude response(band pass filter)');
subplot(2,1,2);
plot(w/pi,p);
grid;
xlabel('normalized frequency-->');
ylabel('phase in radians-->');
title('phase response(band pass filter)');
```

PROGRAM:

```
clc;
clear all;
close all;
ap=0.5;
as=50;
fp=1000;
fs=2000;
f=5000;
wp=2*fp/f;
ws=2*fs/f;
wn=[wp,ws];
[n]=buttord(wp,ws,ap,as);
[b,a]=butter(n,wn,'stop');
w=0:0.01:pi;
h=freqz(b,a,w);
p=angle(h);
mag=20*log10(abs(h));
subplot(2,1,1);
plot(w/pi,mag);
grid;
xlabel('normalized frequency');
ylabel('magnitude in db-->');
title('magnitude response(band stop filter)');
subplot(2,1,2);
plot(w/pi,p);
grid;
xlabel('normalized frequency-->');
ylabel('phase in radians-->');
title('phase response(bandstop filter)');
```

PROGRAM:

```
clc;
close all;clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;w2=2*ws/fs;
[n,wn]=cheb1ord(w1,w2,rp,rs,'s');
[b,a]=cheby1(n,rp,wn,'s');
W=0:.01:pi;
[h,om]=freqs(b,a,W);
M=20*log10(abs(h));
An=angle(h);
subplot(2,1,1);
plot(om/pi,M);
ylabel('Gain in dB --.');
xlabel('(a) Normalised frequency --.');
subplot(2,1,2);
plot(om/pi,An);
xlabel('b) Normalised frequency --.');
ylabel('Phase in radians --.');
```

PROGRAM:

```
clc;
close all;clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=cheb2ord(w1,w2,rp,rs,'s');
[b,a]=cheby2(n,rs,wn,'high','s');
w=0:.01:pi;
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('Gain in dB --.');
xlabel('(a) Normalised frequency --.');
subplot(2,1,2);
plot(om/pi,an);
xlabel('(b) Normalised frequency --.');
ylabel('Phase in radians --.');
```

DOWN SAMPLING

PROGRAM:

```
clc;
clear all;
close all;
N=input('Enter the sequence length N:');
M=input('Enter the down sampling factor M:');
f1=0.05;
f2=0.2;
t=0:1:N-1;
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);
x1=x(1:M:N);
t1=1:1:N/M;
subplot(3,1,1);
plot(t,x);
xlabel('time-->');
ylabel('amplitude-->');
title('Input analog signal');
subplot(3,1,2);
stem(t,x);
xlabel('n-->');
ylabel('amplitude-->');
title('Sampled signal');
subplot(3,1,3);
stem(t1-1,x1);
xlabel('n-->');
ylabel('amplitude==>');
title('Down sampled signal');
```

UPSAMPLING

PROGRAM:

```
clc;
clear all;
close all;
N=input('Enter the sequence length N:');
L=input('Enter the upsampling factor L:');
f1=0.01;
f2=0.2;
t=0:1:N-1;
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);
x1=[zeros(1,L*N)];
t1=1:1:L*N;
j=1:L:L*N;
x1(j)=x;
subplot(3,1,1);
plot(t,x);
xlabel('time-->');
ylabel('amplitude-->');
title('Input analog signal');
subplot(3,1,2);
stem(t,x);
xlabel('n-->');
ylabel('amplitude-->');
title('sampled signal');
subplot(3,1,3);
stem(t1-1,x1);
xlabel('n-->');
ylabel('amplitude==>');
title('upsampled signal');
```

SAMPLING RATE CONVERTOR BY RATIONAL FACTOR I/D

PROGRAM:

```
clc;
clear all;
close all;    %Sampling rate conversion by factor I/D = 5/4
Fx=20;       %Original sampling frequency in Hz
Tx=1/Fx;      %Original sampling period in seconds
tx=0:Tx:1; % Time vector tx
x=0.7*sin(2*pi*tx); % Original sequence
y=resample(x,5,4);  % Re-sampling by rational factor I/D
ty=(0:(length(y)-1))*4*Tx/5;  % New time vector ty
figure(1)
stem(tx,x,'*')
hold on
stem(ty,y,'-.r')
title('Original sequence and Sampling rate conversion by
factor I/D')
legend('Original sequence','Resampled signal by rational
factor I/D')
xlabel('Time (s)'),
ylabel('Amplitude'),
axis([0,1,-1,1])
```

PROGRAM:

```
INP1    .SET 0H
INP2    .SET 1H
OUT     .SET 2H
.mmregs
.text
START:
        LD  #140H,DP
        RSBX  CPL
        NOP
        NOP
        NOP
        NOP
        LD      INP1,A
        ADD   INP2,A
        STL     A,OUT
HLT:  B HLT
```

PROGRAM:

```
INP1 .SET 0H
INP2 .SET 1H
OUT .SET 2H
.mmregs
.text
START:
        LD #140H,DP
        RSBX CPL
        NOP
        NOP
        NOP
        NOP
        LD INP1,A
        SUB INP2,A
        STL A,OUT
HLT:   B HLT
```

PROGRAM:

```
DIVID .SET 0H
DIVIS .SET 1H
QOUT .SET 2H
REMAIN .SET 3H
.mmregs
.text
START:
        STM #140H,ST0
        RSBX CPL
        RSBX FRCT
        NOP
        NOP
        NOP
        NOP
        LD DIVID,A
        RPT #0FH
        SUBC DIVIS,A
        STL A,QOUT
        STH A,REMAIN
HLT:   B HLT
```

PROGRAM:

```
.mmregs
.text
START:
        LD #00H,A
        STM #1000H,AR4
        STM #2000H,AR5
        STM #3000H,AR6
        LD *AR4,A
        ADD *AR5,A
        STL A,*AR6+
HLT: B HLT
```

PROGRAM:

```
.mmregs
.text
START:
        LD #00H,A
        STM #1000H,AR4
        STM #2000H,AR5
        STM #3000H,AR6
        LD *AR4,A
        SUB *AR5,A
        STL A,*AR6+
HLT:   B HLT
```

PROGRAM:

```
.mmregs
.text
START:
        STM #0140H,ST0
        STM #40H,PMST
        STM #0A000H,AR0
        ST #1H,*AR0
        LD *AR0+,T
        ST #2H,*AR0
        MPY *AR0+,A
        STL A,*AR0
HLT:    B HLT
.END
```

PROGRAM:

To Generate Square Wave

```
ioport int port4;
main()
{
int i,x;
while(1)
{
for(i=0;i<=0x5ff;i++)
{
x=0x0000;
port4=x;
}
for(i=0;i<=0x5ff;i++)
{
x=0xfff;
port4=x;
}
}
}
```

To Generate Saw Tooth Wave

```
ioport int port4;
main()
{
int i,x;
while(1)
{
x=0x0;
for(i=0;i<=0xfff;i++)
{
port4 = x;
x=x+5;
}
}
}
```

To Generate Triangle Wave

```
ioport int port4;
main()
{
int i,x;
while(1)
{
x=0x000;
for(i=0;i <= 0xfff;i++)
{
port4 = x;
x=x+1;
}
for(i=0;i <= 0xfff;i++)
{
port4 = x;
x=x-1;
}
}
}
```