```
% DISCRETE TIME CONVOLUTION

%{
%1)
x=input("Enter vector x: ");
h=input("Enter vector h: ");
N1=input("Enter the value of N1: ");
N2=input("Enter the value of N2: ");
[v,N]=Convolution(x,h,N1,N2);
disp(v);
fprintf("Reference is at %d.\n",N);
%}

%2)
%a)
subplot(241);
x=[1 2 3];
h=[1 -1];
N1=1;
N2=1;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(245);
[v,N]=Convolution(x,h,N1,N2);
nf=zeros(1,n+m-1);
j=1;
```

```
for i=1-N:-N+n+m-1
    nf(1,j)=i;
    j=j+1;
end
stem(nf,v,'Color','g','LineWidth',2);

%b)
subplot(242);
x=[1 2 3];
h=[1 -1];
N1=2;
N2=1;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(246);
[v,N]=Convolution(x,h,N1,N2);
nf=zeros(1,n+m-1);
j=1;
for i=1-N:-N+n+m-1
    nf(1,j)=i;
    j=j+1;
end
stem(nf,v,'Color','g','LineWidth',2);

%c)
subplot(243);
x=[1 2 3];
h=[1 -1];
```

```
N1=3;
N2=2;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(247);
[v,N]=Convolution(x,h,N1,N2);
nf=zeros(1,n+m-1);
j=1;
for i=1-N:-N+n+m-1
    nf(1,j)=i;
    j=j+1;
end
stem(nf,v,'Color','g','LineWidth',2);

%d)
subplot(244);
x=[0.5 1];
h=[2 0 2];
N1=2;
N2=3;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(248);
[v,N]=Convolution(x,h,N1,N2);
nf=zeros(1,n+m-1);
j=1;
for i=1-N:-N+n+m-1
    nf(1,j)=i;
    j=j+1;
end
stem(nf,v,'Color','g','LineWidth',2);
% CONVOLUTION BUILT-IN FUNCTION

%3)
help conv;

%{
%4)
x=input("Enter vector x: ");
h=input("Enter vector h: ");
N1=input("Enter the value of N1: ");
N2=input("Enter the value of N2: ");

v=conv(x,h);
disp(v);

%}

%5)
%{
Explanation:- In 2(b) delaying the input signal delays the output signal
This the due to the nature/basic definition of convolution.
Convolution of x and h is defined as:-
Sigma[ k from -INF to +INF (x[k]h[n-k]) ]
Here, we observe that delaying/shifting x and keeping h constant will
result in shifting/delaying all the terms of the summation from -INF to
```

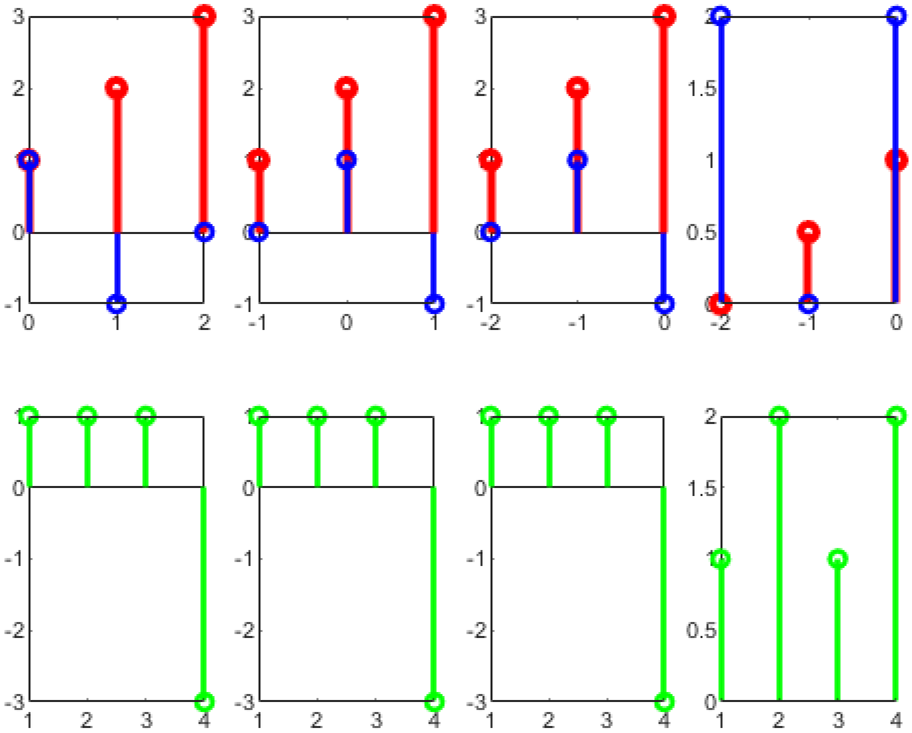
```
+INF. Hence, due to this we get a delayed output convolution y[n]
%}

%6)
help convmtx;

%7)
%a)
subplot(241);
x=[1 2 3];
h=[1 -1];
N1=1;
N2=1;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(245);
%v=zeros(1,n+m-1);

A=convmtx(h,length(x));
con=x*A;
stem(con,'Color','g','LineWidth',2);

%b)
subplot(242);
x=[1 2 3];
h=[1 -1];
N1=2;
N2=1;
```

```
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(246);
A=convmtx(h,length(x));
con=x*A;
stem(con,'Color','g','LineWidth',2);

%c)
subplot(243);
x=[1 2 3];
h=[1 -1];
N1=3;
N2=2;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
```

```
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(247);
A=convmtx(h,length(x));
con=x*A;
stem(con,'Color','g','LineWidth',2);

%d)
subplot(244);
x=[0.5 1];
h=[2 0 2];
N1=2;
N2=3;
h=h';
n=length(x);
m=length(h);
nfinal=max(N1,N2)+max(n-N1,m-N2);
nx=zeros(1,nfinal);
j=1-max(N1,N2);
for i=1:nfinal
    nx(1,i)=j;
    j=j+1;
end

X=zeros(1,nfinal);
H=zeros(1,nfinal);
h=h';
j=1;
for i=max(N1,N2)-N1+1:max(N1,N2)-N1+n
    X(1,i)=x(1,j);
    j=j+1;
end
j=1;
for i=max(N1,N2)-N2+1:max(N1,N2)-N2+m
    H(1,i)=h(1,j);
    j=j+1;
end

stem(nx,X,'Color','r','LineWidth',3); hold ON;
stem(nx,H,'Color','b','LineWidth',2); hold OFF;

subplot(248);
A=convmtx(h,length(x));
```

```
con=x*A;
stem(con,'Color','g','LineWidth',2);
```

```
 CONV Convolution and polynomial multiplication.
    C = CONV(A, B) convolves vectors A and B.  The resulting vector is
    length MAX([LENGTH(A)+LENGTH(B)-1,LENGTH(A),LENGTH(B)]). If A and B are
    vectors of polynomial coefficients, convolving them is equivalent to
    multiplying the two polynomials.

    C = CONV(A, B, SHAPE) returns a subsection of the convolution with size
    specified by SHAPE:
      'full'  - (default) returns the full convolution,
      'same'  - returns the central part of the convolution
                that is the same size as A.
      'valid' - returns only those parts of the convolution
                that are computed without the zero-padded edges.
                LENGTH(C)is MAX(LENGTH(A)-MAX(0,LENGTH(B)-1),0).

    Class support for inputs A,B:
       float: double, single

    See also DECONV, CONV2, CONVN, FILTER, XCORR, CONVMTX.

    Note: CONVMTX is in the Signal Processing Toolbox.

    Documentation for conv
       doc conv

    Other uses of conv

       tall/conv

 CONVMTX Convolution matrix.
    A = CONVMTX(H,N) returns the convolution matrix, A, such that
    convolution of H and another vector, X, of length N with the same
    vector orientation as H may be expressed by matrix multiplication:
    When H and X are column vectors, A*X is the same as CONV(H,X);
    when H and X are row vectors, X*A is the same as CONV(H,X).

    % Example:
    %   Generate a simple convolution matrix.

    h = [1 2 3 2 1];
    convmtx(h,7)        % Convolution matrix

    See also CONV.

    Documentation for convmtx
       doc convmtx

    Other uses of convmtx

       signal/convmtx
```



Published with MATLAB® R2023a