LAB – 3: Linear Convolution and Circular Convolution

Name: K Sri Rama Rathan Reddy - 2022102072

Team Mate: B Karthikeya - 2022102042

Team Name: Noicifiers

QUESTION – 3.1.a:

CONVOLUTION FUNCTION:

```
function y = convolution(x,h)
   len_x = length(x);
   len_h = length(h);
   y = zeros(1,(len_x + len_h - 1));
   for a = 1:length(y)
       for b = max(1, a+1-len_h): min(a, len_x)
           y(a) = y(a) + x(b)*h(a-b+1);
       end
   end
end
```

EXPLANATION:

Convolution

The convolution of two vectors, u and v, represents the area of overlap under the points as v slides across u. Algebraically, convolution is the same operation as multiplying polynomials whose coefficients are the elements of u and v.

Let m = length(u) and n = length(v) . Then w is the vector of length m+n-1 whose kth element is

$$w(k) = \sum_j u(j)v(k - j + 1).$$

The sum is over all the values of j that lead to legal subscripts for u(j) and v(k-j+1), specifically j = max(1,k+1-n):1:min(k,m). When m = n, this gives

```
w(1) = u(1)*v(1)
w(2) = u(1)*v(2)+u(2)*v(1)
w(3) = u(1)*v(3)+u(2)*v(2)+u(3)*v(1)
...
w(n) = u(1)*v(n)+u(2)*v(n-1)+ ... +u(n)*v(1)
...
w(2*n-1) = u(n)*v(n)
```

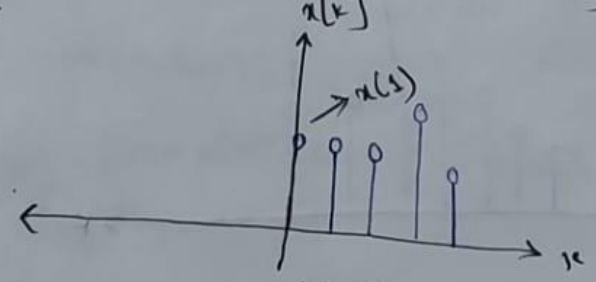
* If we convolve a signal ~~A~~ $x[n]$ of length len_x and signal $h[n]$ with length len_h.

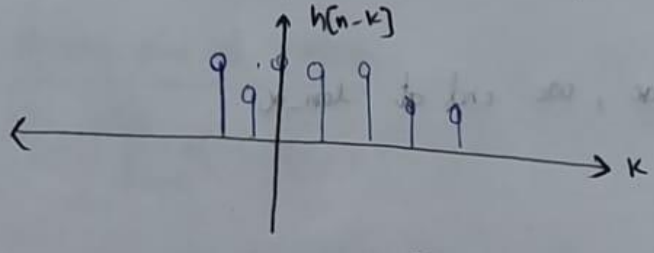
$$y = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = x[n] * h[n]$$

length of y = len_x + len_~~y~~ − 1

* For writing the function, We need two for loops, One for iterating through 'y' array and store values in it Second loop for calculating the value to be stored in y array.

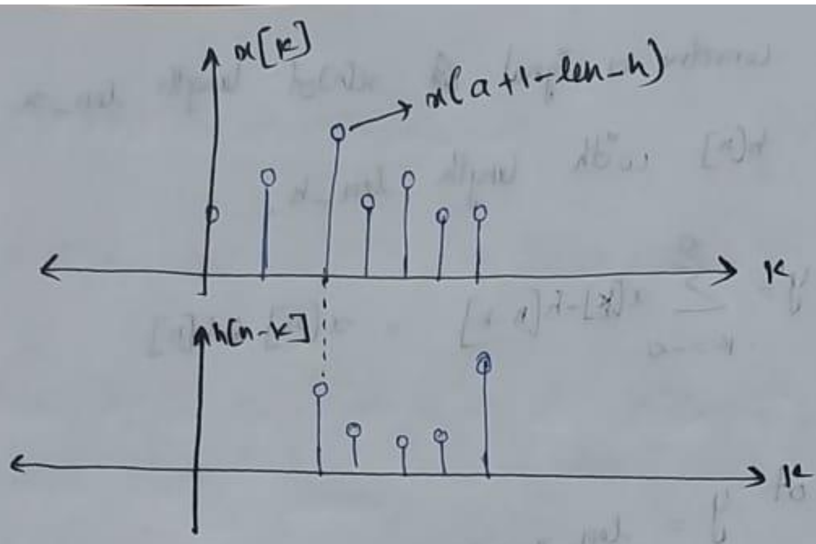
* loop-1 from 1 to length of y
* loop-2 from : max (1, a+1−len_h) to min (a, len_x)
*





in this case, we start ~~the~~ second loop from 1

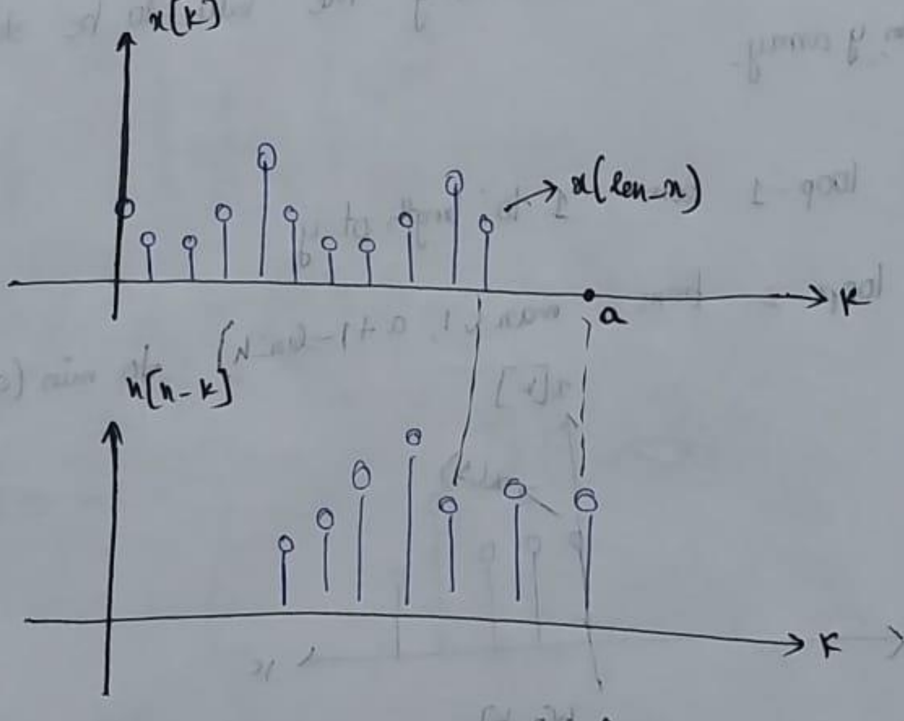
*



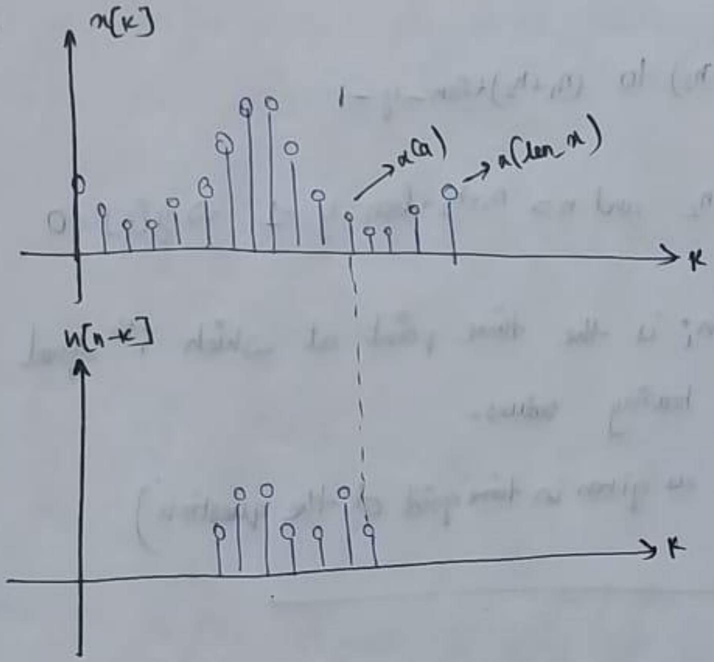
in this case, we start from a+1-len_h

∴ We write max(1, a+1-len_h).

*



in this case, we end at len_x



$\therefore$ in this case, we end at a

$\therefore$ We write min(a, len_x)

* Considering the 4-cases of overlap between x and h, we iterate the 2nd loop from max(1, a+1-len_h) to min(a, len_x)

y(a) = y(a) + x(b)* h(a-b+1) accounts for the summation value.

* <u>Starting time of plotting:-</u>

$$y = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

n-k = starting time of 2nd signal
↳ starting time of 1st signal

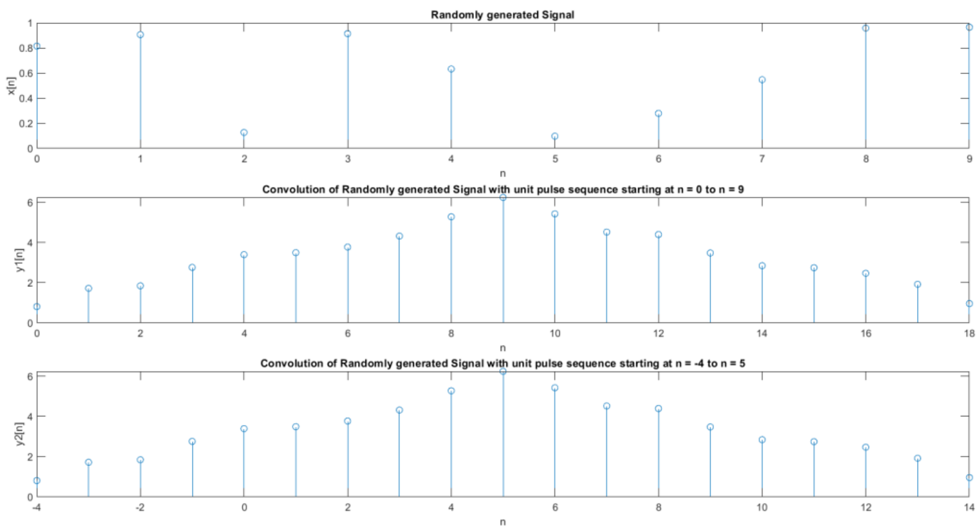
else all will be zeros before

$n : (n_1+n_2)$ to $(n_1+n_2)+len\_y-1$

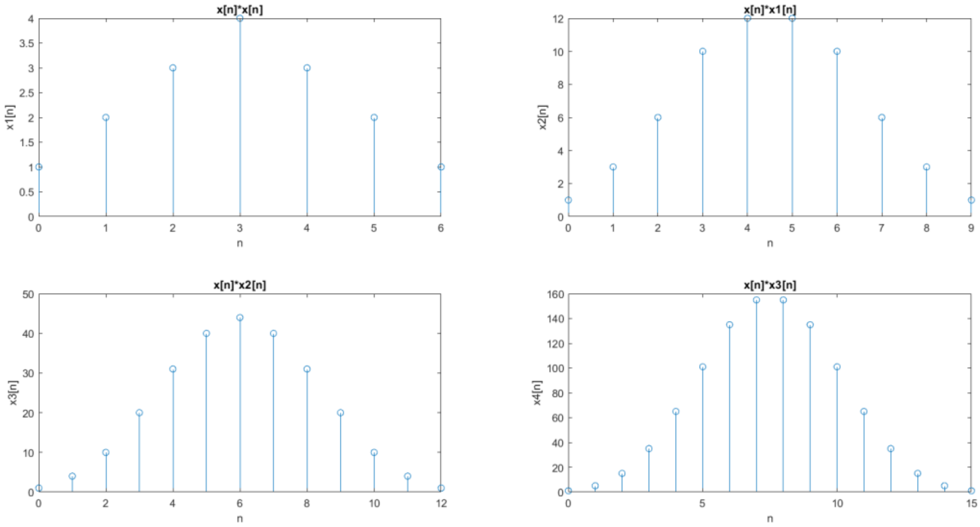
$n < n_1+n_2$ and $n > n_1+n_2+len\_y-1 \Rightarrow y[n]=0$

where n_i is the time point at which i^{th} signal starts having values.
(i.e, as given in time grid of the question)

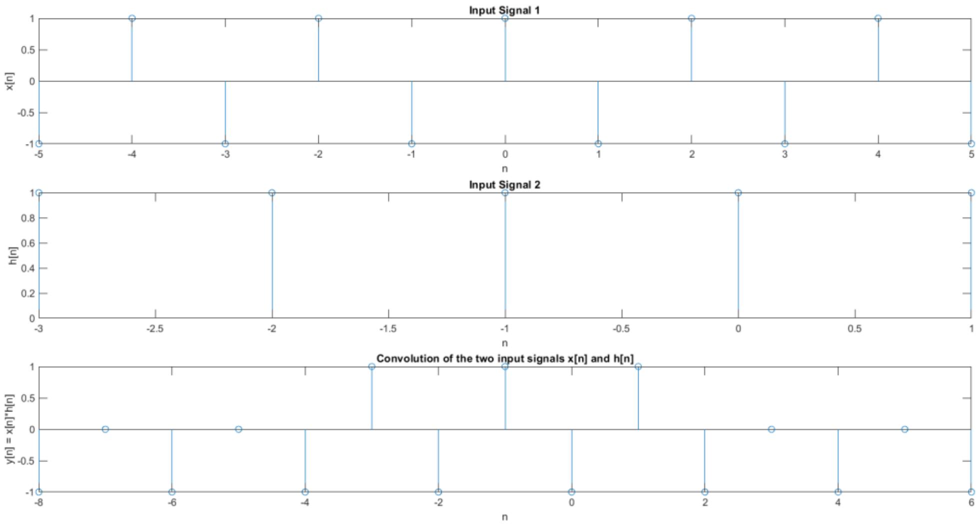
QUESTION – 3.1.a:



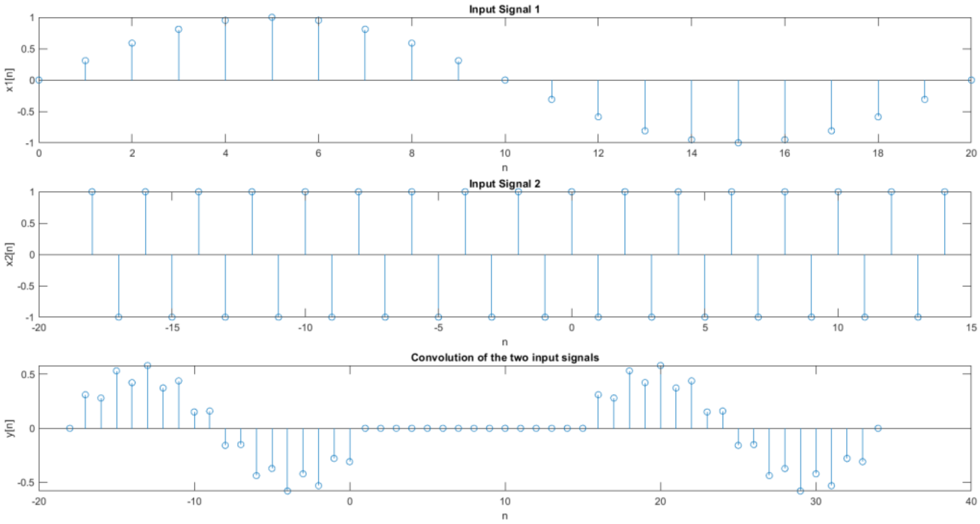
QUESTION – 3.1.b:



QUESTION – 3.1.c:



QUESTION – 3.1.d:



QUESTION – 3.2.a:

```
% Generation of Finite Length Sequences:
n = 0:1:9;
x1 = randn(1,10); %Random Gaussian sequence of length 10
x2 = [0 0 0 1 zeros(1,6)]; %first 10 samples of the signal δ[n − 3] starting from
n = 0
```

QUESTION – 3.2.b:

```
% Linear and circular Convolutions using inbuilt functions:
y_lin = conv(x1,x2); % Linear Convolution
n1 = 0:length(y_lin)-1;
y_cir = cconv(x1,x2); % circular Convolution
n2 = 0:length(y_cir)-1;
```

QUESTION – 3.2.c: