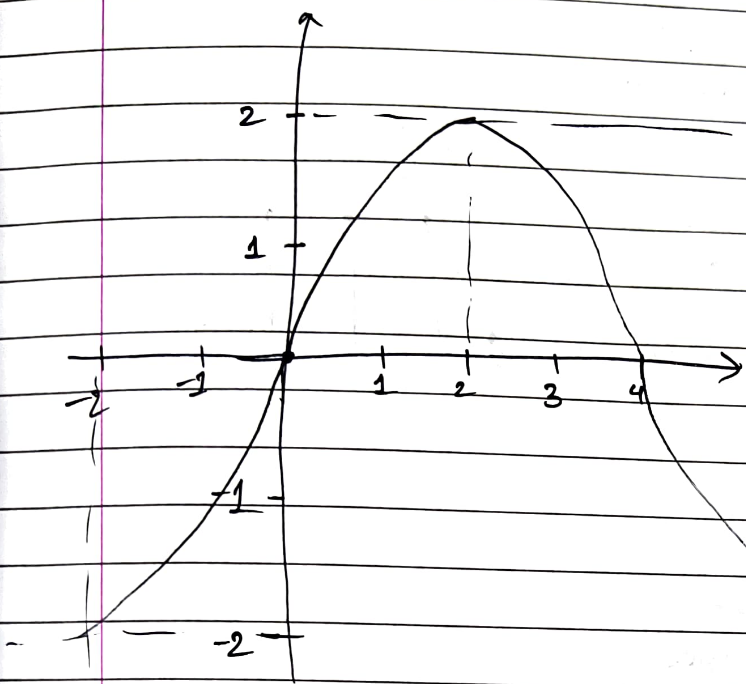
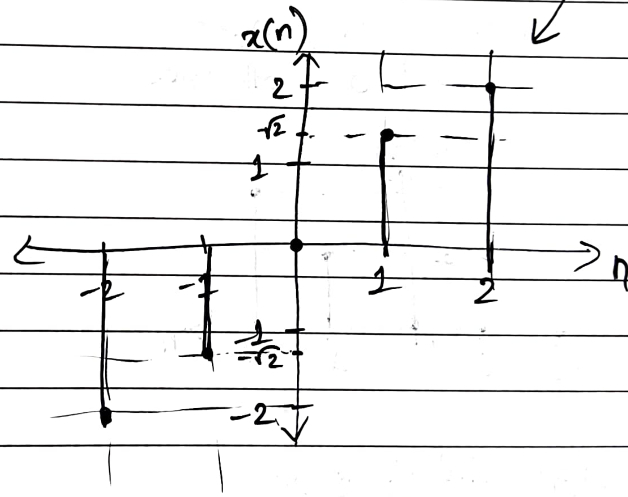
DSIP

$x(n) = 2\sin\left(\frac{\pi}{4}n\right)$



But this is continuous

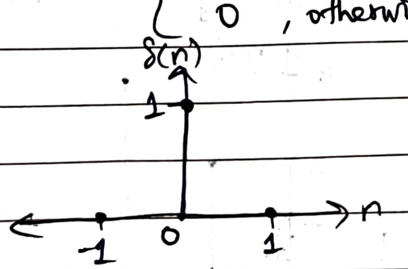
Discrete needed for discrete signal



→ <u>Standard Signals</u> :

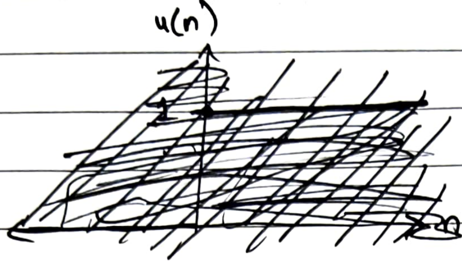
(1) <u>Unit Impulse</u> ↓

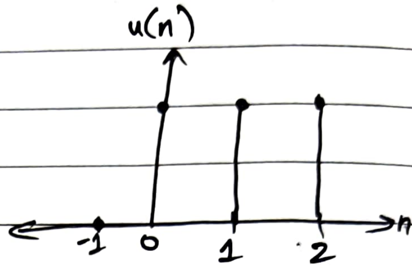
$$\delta(n) = \begin{cases} 1 & , n = 0 \\ 0 & , \text{otherwise} \end{cases}$$



(2) Unit Step ⤵

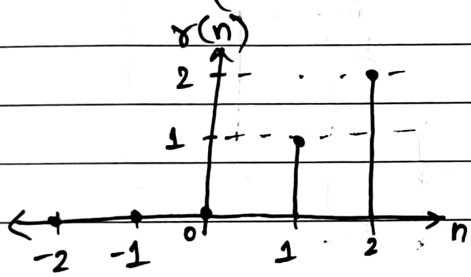
$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$





(3) Unit Ramp ⤵

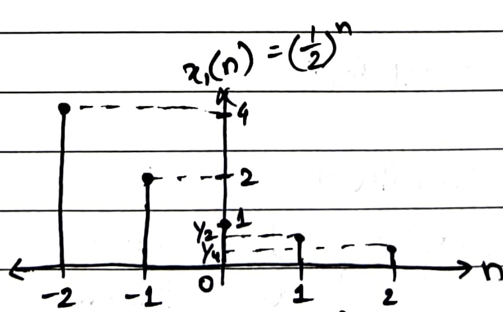
$$r(n) = \begin{cases} n, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



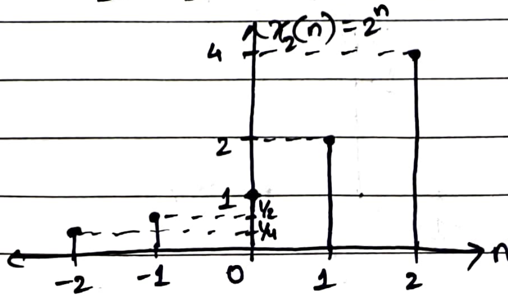
(4) Exponential Function ⤵

$x(n) = a^n$

$x_1(n) = \left(\frac{1}{2}\right)^n$



$x_2(n) = 2^n$



⇒ ~~Exa~~ If $a = re^{j\theta}$ then $x(n) = r^n e^{j\theta n}$ is complex exponential

$\therefore x(n) = r^n[\cos(\theta n) + j\sin(\theta n)]$ → Magnitud $= r^n$
↳ Phase $= \theta n$

→ Operations on dependent variable

↳ Amplitude Scaling
e.g. $y(n) = A \cdot x(n)$

↳ Addition Scaling
e.g. $y(n) = x_1(n) + x_2(n)$

↳ Multiplication Scaling
e.g. $y(n) = x_1(n) \times x_2(n)$

→ Operations on independent variable

↳ Time Delay
e.g. $y(n) = x(n - n_0)$
If $n_0 = +ve$ → shifted to right
if $n_0 = -ve$ → shifted to left

↳ Folding
e.g. $y(n) = x(-n)$

↳ Time Scaling
e.g. $y(n) = x(an)$
If $a \geq 1$ → downsampling
if $0 < a < 1$ → upsampling

→ Classification of Signals:

(1) Deterministic / Random

(2) Periodic / Non-periodic

↳ Periodic :-

$x(t) = x(t+T)$

$x(n) = x(n+N)$

→ $x(t) = \cancel{A \sin}\ A \sin(\Omega t) = A \sin(2\pi f t)$ NOTE

Ω ↓ rad/s ; f ↓ cycles/s

($t = nT$ ← we know this)

[T = sampling interval , F = sampling frequency]

→ To be periodic, $x(n) = \sin(\omega n) = \sin(\omega(n+N))$

$= \sin(\omega n)\cos(\omega N) + \cos(\omega n)\sin(\omega N)$

To be true, $\cos(\omega N) = 1$ and $\sin(\omega N) = 0$

$\therefore \omega N = 2\pi k \quad \therefore \omega = \frac{2\pi k}{N} \quad f = \frac{k}{N}$

E.g. $\sin\left(\frac{3\pi}{4} n\right) \rightarrow [k = 3, N = 8]$

k → how many analog signals are required to show periodicity

N → how many samples are present in one period of digital domain

(3) Energy / Power

↳ Energy $E_x = \sum_{-\infty}^{\infty} [x(n)]^2$

If E_x is finite, signal is an energy signal

If E_x is infinite, it will be a power signal

↳ Power $P_x = \frac{1}{N} \sum_{n=0}^{N-1} [x(n)]^2$

(4) Even / Odd

↳ If $x(n) = x(-n) \Rightarrow$ Even signal

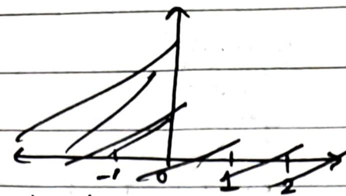
↳ If $x(n) = -x(-n) \Rightarrow$ odd signal

↳ Any arbitrary signal can be represented as sum of even part of the signal and odd part of the signal: $x(n) = x_e(n) + x_o(n)$

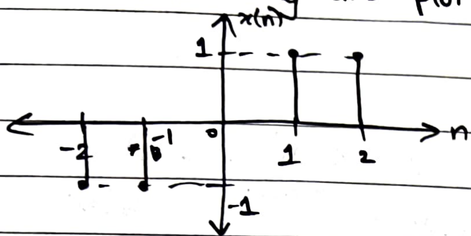
Also, $x(-n) = x_e(-n) + x_o(-n) = \cancel{x_e(-n)} = x_e(n) - x_o(n)$

$\therefore 2x_e(n) = x(n) + x(-n) \rightarrow x_e(n) = \frac{x(n) + x(-n)}{2}$; $x_o(n) = \frac{x(n) - x(-n)}{2}$

Q. $x(n) = \begin{cases} 1 & ; n = 1, 2 \\ -1 & ; x = -1, -2 \\ 0 & ; \text{otherwise} \end{cases}$

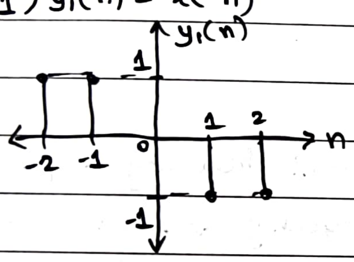


Obtain the following and plot the signal

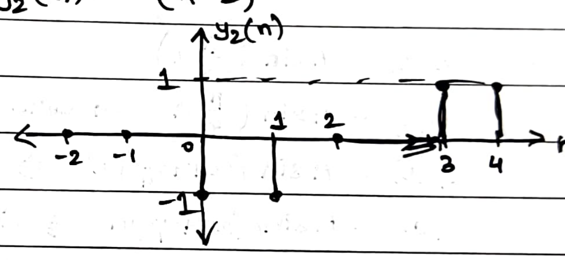


(1) $y_1(n) = x(-n)$
(2) $y_2(n) = x(n-2)$
(3) $y_3(n) = x(n+3)$
(4) $y_4(n) = x(-n+1)$
(5) $y_5(n) = x(2n)$
(6) $y_6(n) = x(\frac{n}{2})$
(7) $y_7(n) = x(2n+3)$
(8) $y_8(n) = x(n)[u(n) - u(n-1)]$
(9) $y_9(n) = x(n)[u(-n)]$
(10) $x_e(n)$ & $x_o(n)$

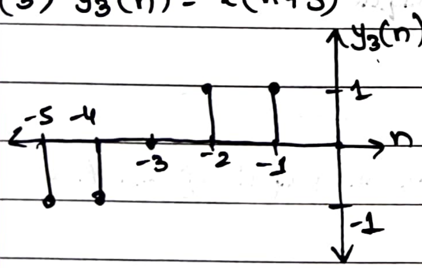
Ans. (1) $y_1(n) = x(-n)$



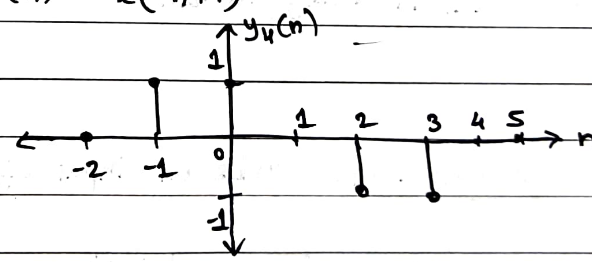
(2) $y_2(n) = x(n-2)$



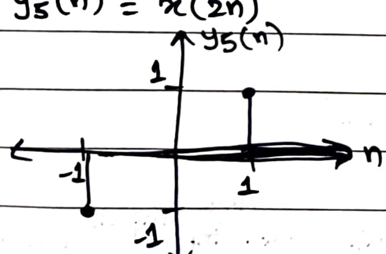
(3) $y_3(n) = x(n+3)$



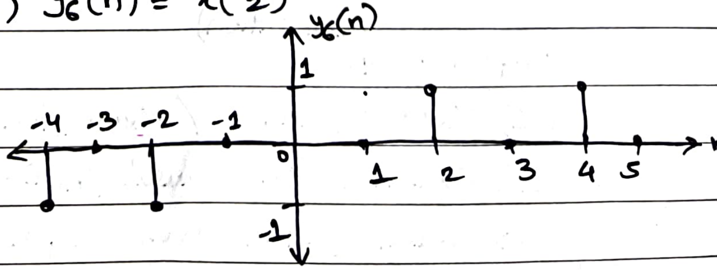
(4) $y_4(n) = x(-n+1)$



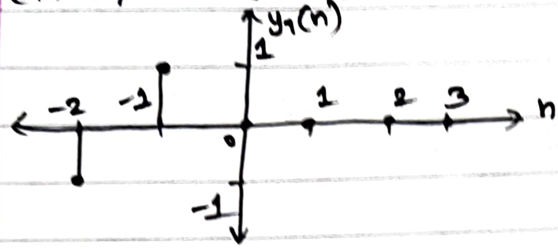
(5) $y_5(n) = x(2n)$



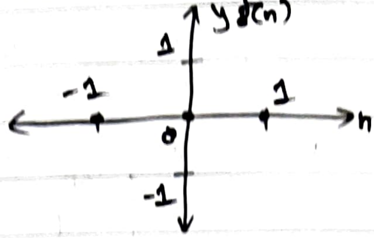
(6) $y_6(n) = x(\frac{n}{2})$



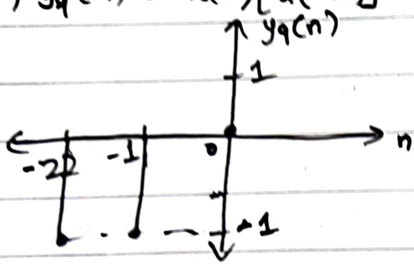
(7) $y_7(n) = x(2n+3)$



(8) $y_8(n) = x(n)\left[u(n) - u(n-1)\right]$

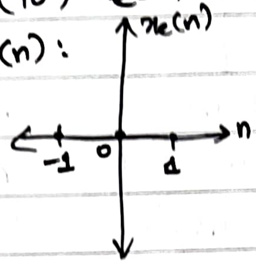


(9) $y_9(n) = x(n)\left[u(-n)\right]$

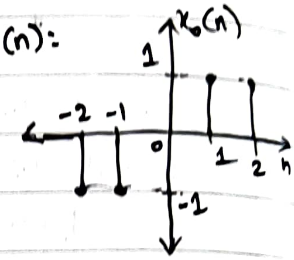


(10) $x_e(n) + x_o(n)$

$x_e(n)$:



$x_o(n)$:



→ $x(t) = A\sin\left(\frac{\pi}{4}t\right)$

$x(n) = A\sin\left(\frac{\pi}{4}n\right)$ → considering $T = 1s$.

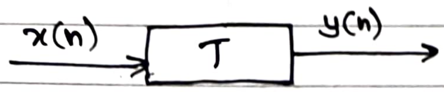
$x_1(t) = A\sin\left(\left(2\pi + \frac{\pi}{4}\right)t\right)$ ⇒ will have different frequency compared to $x(t)$.

$x_1(n) = A\sin\left(\left(2\pi + \frac{\pi}{4}\right)n\right)$ ⇒ will have same frequency compared to $x(n)$.

So in digital domain : $-\pi$ to π is unique (frequency)

but in analog domain : $-\infty$ to ∞ is unique

→ System: works on input signal to obtain an output signal. Acts as a filter.



Properties (Classification)

(1) Memory (static) / Memoryless (dynamic)

(2) Causal / Non-causal / Anti-causal

↳ Causal system works on present inputs, ~~past inp~~ previous inputs and previous outputs.

↳ Non-causal system works on next (future) inputs as well

↳ Anti-causal system would not have any causal terms.

(3) Linear / Non-linear

↳ Linear : $2 \rightarrow \boxed{L} \rightarrow 4$, $4 \rightarrow \boxed{L} \rightarrow 8$, $2+4 \rightarrow \boxed{L} \rightarrow 4+8$

Additivity and homogenuity features } → Superposition Theorem

(4) ~~Time~~ Time-invariant / Time-variant

↳ Time-invariant systems do not vary with time.

↳ Time-variant systems may change their behaviour with time.

(5) Stability

↳ Bounded Input - Bounded output (BIBO)

↳ If both input and output of a system is bounded, it is called a stable system.

Q. $y(n) = x(n-1)$. Classify this system.

Ans. → Memory (static) system since it needs to store previous input.

→ Causal system, since it uses only previous inputs.

→ Linear system, since linear transformations.

→ Time-invariant, since system behaviour does not change with time.

Q. $y(n) = [x(n)]^2$. Classify the system.

Ans. → Memoryless (dynamic) system, since only current input.

→ Causal system

→ Non-linear system, since non-linear transformations.

→ Time-invariant

→ Unstable (since squares could become very large)

$\boxed{\text{NOTE: LTI = Linear Time-Invariant system}}$

⇒ Impulse Response :

It is the response (output) of a system when it is provided impulse input (see 'Unit Impulse') $\left[\delta(n) = \begin{cases} 1, & n=0 \\ 0, & \text{otherwise} \end{cases} \right]$

$x(n) = \delta(n) \rightarrow \boxed{LTI} \rightarrow y(n) = h(n)$ [Impulse Response]

→ we know we can write $x(n)$ as:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

∴ Using linearity and time-invariance properties,

If input is $x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$

then output is $y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$ //

This is known as Convolution Theorem // $\boxed{[x(n) * h(n)] = y(n)}$

Q. $x(n) = [0, \underset{\uparrow}{2}, 3, -1]$, $h(n) = [-1, \underset{\uparrow}{-3}, 2]$

Use Convolution Theorem to solve.

Ans. We know,

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$$

We can write $x(n)$ as $x(n) = 2\,\delta(n) + 3\,\delta(n-1) - 1\,\delta(n-2)$

Now taking inputs one at a time,

$2\,\delta(n) \longrightarrow [-2, \underset{\uparrow}{-6}, 4]$

$3\,\delta(n-1) \longrightarrow [0, \underset{\uparrow}{-3}, -9, 6]$

$-1\,\delta(n-2) \longrightarrow [0, \underset{\uparrow}{0}, 1, 3, -2]$

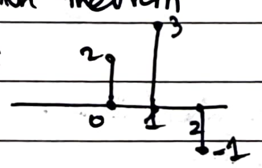
Now, $y(n)$ will be adding up all these,

∴ $y(n) = [-2, \underset{\uparrow}{-9}, -4, 9, -2]$ //

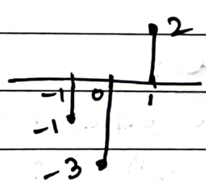
← This was using basic definition of linearity and time-invariance

By convolution theorem

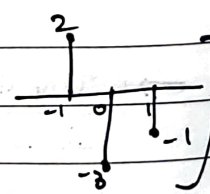
$x(k)$:



$h(n)$:



$h(-k)$:



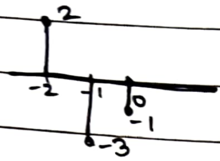
∴ $y(0) = \sum_{k=-\infty}^{\infty} x(k)\,h(-k)$

∴ From this, $y(0) = (0 \times 2) + (2 \times -3) + (3 \times -1) + (-1 \times 0)$

$= -6 - 3 = -9$ //

$y(-1) = \sum_{k=-\infty}^{\infty} x(k)\, h(-1-k)$

$h(-1-k)$:

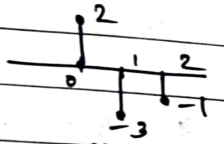


from this, $\cancel{y(0)}\; y(-1) = (0\times 2) + (0\times -3) + (-1\times 2) + (3\times 0) + (-1\times 0)$

$= -2$

Similarly, $y(1) = \sum_{k=-\infty}^{\infty} x(k)\, h(1-k)$

$h(1-k)$:

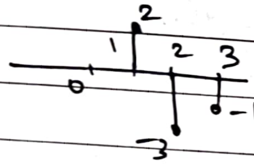


From this, $y(1) = (2\times 2) + (3\times -3) + (-1\times -1)$

$= 4 - 9 + 1 = -4$

Also, $y(2) = \sum_{k=-\infty}^{\infty} x(k)\, h(2-k)$

$h(2-k)$:

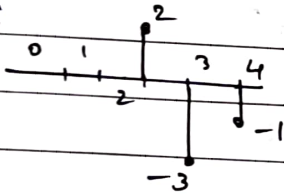


from this, $y(2) = (2\times 0) + (3\times 2) + (-1\times -3) + (0\times -1)$

$= 6 + 3 = 9$

Finally, $y(3) = \sum_{k=-\infty}^{\infty} x(k)\, h(3-k)$

$h(3-k)$:



From this, $y(3) = (2\times 0) + (3\times 0) + (-1\times 2) + (0\times -3) + (0\times -1)$

$= -2$

← Convolution Theorem

Also, ~~tabular~~ matrix method ↓

	2 ($n=0$)	3	-1
-1	-2	-3	1
-3 ($n=0$)	6	-9	3
2	4	6	-2

Add diagonally to get answer!

NOTE: Basic definition and tabular methods are only possible for simple problems like these.

- Convolution Theorem Properties :

→ Commutative Property ↓

$$x(n) * h(n) = h(n) * x(n)$$

→ Associative Property ↓

$$x(n) * h_1(n) * h_2(n) = x(n) * h_2(n) * h_1(n)$$

→ Distributive Property ↓

$$[x(n) * h_1(n)] + [x(n) * h_2(n)] = x(n) * [h_1(n) + h_2(n)]$$

→ Since $x(n) * h(n) = h(n) * x(n) = y(n)$,

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)\, x(n+k)$$

$$= \dots h(-2)k(n+2) + h(-1)k(n+1) + h(0)k(n) + \dots$$

so for memoryless systems,

$$h(k) = 0 \quad \text{if } k \neq 0$$

for causal systems,

$$h(k) = 0 \quad \text{if } k < 0$$

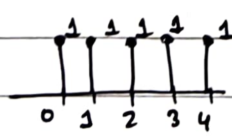
for stable (bounded) systems,

$$\cancel{y(n)}\ \left|\sum_{k=-\infty}^{\infty} h(k)\, x(n-k)\right| \leq \sum_{k=-\infty}^{\infty} |h(k)|\,|x(n-k)|$$

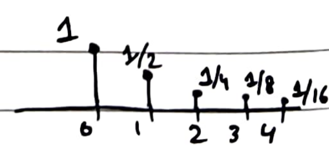
Q. $x(n) = u(n)$, $h(n) = (1/2)^n u(n)$

Find $y(n)$ using convolution.

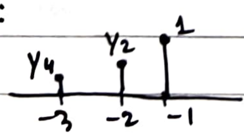
Ans. Plotting :- $x(k)$:



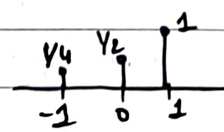
$h(k)$:



$h(-1-k)$:



$h(1-k)$:

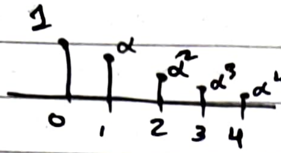


$$\therefore y(n) = \sum_{k=0}^{n} x(k)\, h(n-k) = \sum_{k=0}^{n} 1 \cdot (1/2)^{n-k} = (1/2)^n \sum_{k=0}^{n} 2^k$$

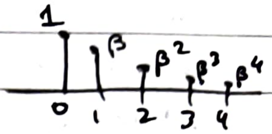
$$= \left(\frac{1}{2}\right)^n \left(\frac{2^n - 1}{2-1}\right)$$

Q. $x(n) = \alpha^n u(n)$ $\quad h(n) = \beta^n u(n)$ $\quad [0 < \alpha, \beta < 1]$

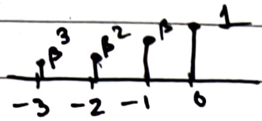
Ans- Plotting:- $x(k)$:



$h(k)$:

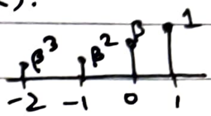


$h(-k)$:



so, similar to last problem.

$h(1-k)$:



$\therefore y(n) = \sum_{k=0}^{n} x(k) h(n-k) = \sum_{k=0}^{n} \alpha^k \beta^{n-k} = \beta^n \sum_{k=0}^{n} \frac{\alpha^k}{\beta^k}$

$$y(n) = \beta^n \cdot \frac{\left(\left(\frac{\alpha}{\beta}\right)^{n+1} - 1\right)}{\left(\frac{\alpha}{\beta}\right) - 1}$$

$= \cancel{\beta^n (1 - 1}$

* Q. $x(n) = (0.5)^n \cdot u(n)$, $h(n) = \delta(n)$

Ans. Since we know property of $\delta(n)$,

$x(n) * \delta(n) = x(n)$

$\therefore y(n) = x(n) = (0.5)^n \cdot u(n)$

Q. $x(n) = [\underset{\uparrow}{1}, 2, -2, 3]$, $h(n) = [\underset{\uparrow}{2}, 1.5, 3]$

Ans. Using tabular method,

x	-3	-2	-1	0	1	2	3	4	5	6
$x(k)$				1	2	-2	3			
$h(k)$				2	1.5	3				
$h(-k)$		3	1.5	2						
$h(-1-k)$	3	1.5	2							
$h(1-k)$			3	1.5	2					
$h(2-k)$				3	1.5	2				
$h(3-k)$					3	1.5	2			
$h(4-k)$						3	1.5	2		
$h(5-k)$							3	1.5	2	

Now, $y(0) = \sum x(k) h(-k) = 2$, $y(1) = \sum x(k) h(1-k) = 1.5 + 4 = 5.5$

Similarly, $y(2) = 2$, $y(3) = 9$, $y(4) = -1.5$, $y(5) = 9$

$\therefore y(n) = [\underset{\uparrow}{2}, 5.5, 2, 9, -1.5, 9]$

⊛ Circular Convolution :

Till now, we learnt linear convolution. ($x(n) \Rightarrow N$, $h(n) \Rightarrow M$, $y(n) \Rightarrow N+M-1$)

Circular convolution is meant for periodic signals.

↳ Periodicity of $y(n)$ will be LCM of periodicities of $x_1(n)$ and $x_2(n)$

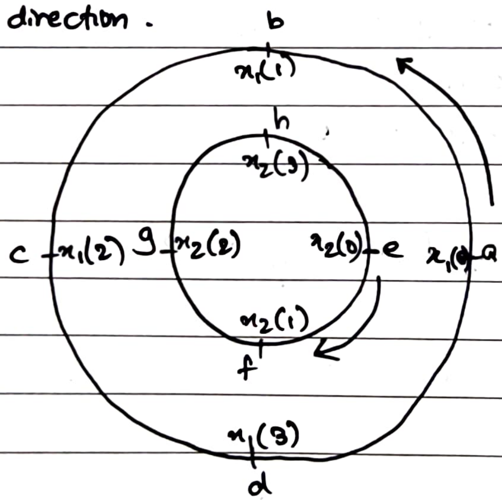
$$\text{if } \underset{P=12}{\underset{\downarrow}{y(n)}} = \underset{N=3}{\underset{\downarrow}{x_1(n)}} \circledast \underset{M=4}{\underset{\downarrow}{x_2(n)}}$$

For circular convolution,

E.g. $x_1(n) = \underbrace{a, b, c, d,}_{N_1=4} a, \ldots$

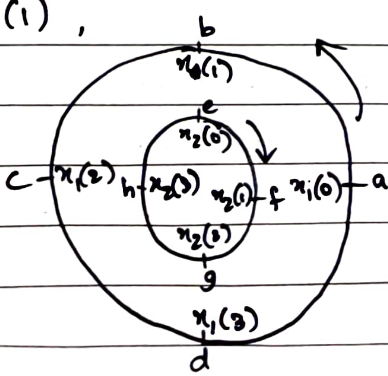
$x_2(n) = \underbrace{e, f, g, h,}_{N_2=4} e, \ldots$

Draw 2 circles one inside other. Outside one will hold values of $x_1(n)$ in anticlockwise direction and inside one holds $x_2(n)$ values in clockwise direction.



$\therefore y(0) = a \cdot e + b \cdot h + c \cdot g + d \cdot f$ //

Similarly, for $y(1)$,



[inner circle rotated anticlockwise one step]

$\therefore y(1) = a \cdot f + b \cdot e + c \cdot h + d \cdot g$ //

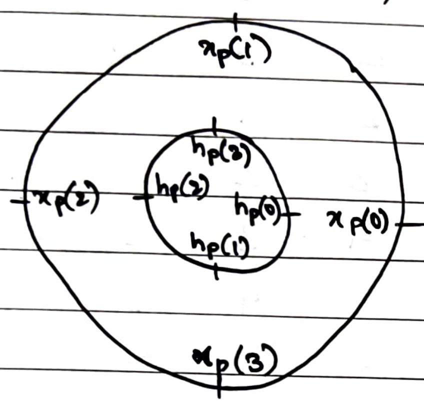
Q. $x_p(n) = \{1, 2, 2, -1\}$

$h_p(n) = \{1, 2, 3, 4\}$

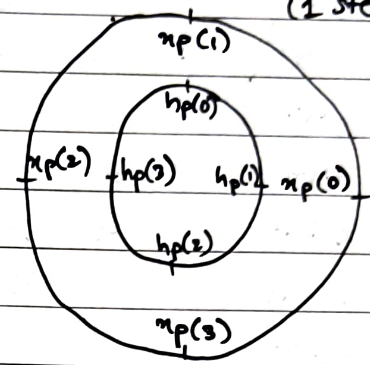
Ans. $y(n) = x_p(n) \circledast h_p(n)$

Since both periods are 4, output period is 4 as well.

(no rotate)



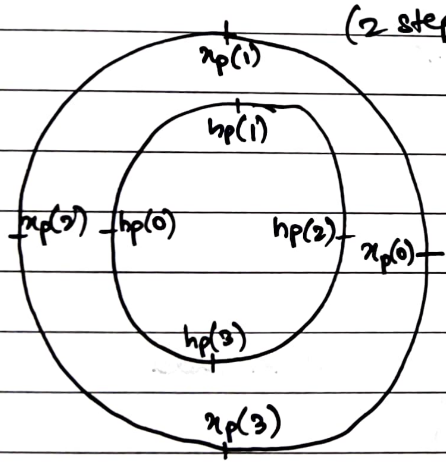
~~(Inner circle~~ (1 step rotated)



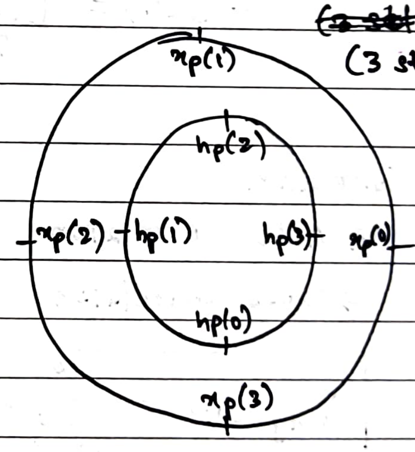
$\therefore y(0) = 1\cdot1 + 2\cdot4 + 2\cdot3 + (-1)\cdot2$
$= 13$

$\therefore y(1) = 1\cdot2 + 2\cdot1 + 2\cdot4 + (-1)\cdot3$
$= 9$

(2 steps rotated)



~~(2 st~~ (3 steps rotated)



$\therefore y(2) = 1\cdot3 + 2\cdot2 + 2\cdot1 + (-1)\cdot4$
$= 5$

$\therefore y(3) = 1\cdot4 + 2\cdot3 + 2\cdot2 + (-1)\cdot1$
$= 13$

$\therefore y(n) = [\underset{\uparrow}{13}, 9, 5, 13]$

⇒ Tabular Method ↓

Q. $x_p(n) = [1, 2, 2, -1]$

$h_p(n) = [1, 2, 3, 4]$

Ans. $y(n) = x_p(n) \circledast h_p(n)$

$x \backslash k$	-3	-2	-1	0	1	2	3	4	5	6
$x(n)$				1	2	2	-1			
$h(-k)$	4	3	2	1	4	3	2	1	4	3
$h(1-k)$		4	3	2	1	4	3	2	1	
$h(2-k)$			4	3	2	1	4	3	2	1
$h(3-k)$				4	3	2	1	4		

↓

from this, $y(0)$ can be found using $x(n)$ and $h(-k)$ rows (and similarly for others)

$y(0) = 13$, $y(1) = 9$, $y(2) = 5$, $y(3) = 13$

$\therefore y(n) = [\underset{\uparrow}{13}, 9, 5, 13]$

* Correlation
  - ↳ Auto-correlation
  - ↳ Cross-correlation

(used in radar)

→ $$\boxed{\gamma_{xy}(m) = \sum_{n=-\infty}^{\infty} x(n)\, y(n-m)}$$

~~the~~ here, m = delay in received signal
n = samples

Also, $$\boxed{\gamma_{yx}(m) = \sum_{n=-\infty}^{\infty} x(n-m)\, y(n)}$$

$$\boxed{\gamma_{xy}(m) = -\gamma_{yx}(m)}$$

initial value $m_1 = n_1 - (n_2 + N_2 - 1)$

final value $m_2 = m_1 + (N_1 + N_2 - 2)$

Q. Find cross-correlation

$x(n) = [\underset{\uparrow}{1}, 1, 2, 2]$ $N_1 = 4$

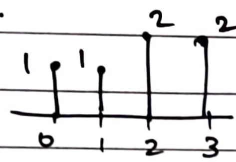
$x_2(n) = [\underset{\uparrow}{1}, 0.5, 1]$ $N_2 = 3$

Ans. Initial value $m_1 = 0 - (0+3-1) = -2$

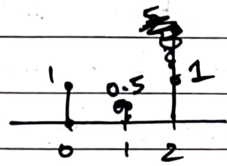
Final value $m_2 = -2 + (4+3-2) = +3$

$$\therefore \gamma_{x_1 x_2}(-2) = \sum_{n=-\infty}^{\infty} x_1(n) \cdot x_2(n-(-2))$$

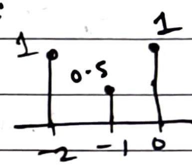
$x_1(n)$:



$x_2(n)$:

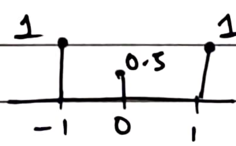


$\therefore x_2(n+2)$:



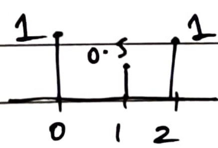
$\therefore \gamma_{x_1 x_2}(-2) = 1 \cdot 1 = 1$

Similarly, $x_2(n+1)$:



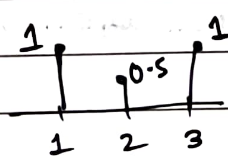
$\therefore \gamma_{x_1 x_2}(-1) = 1 \cdot (0.5) + (1 \cdot 1)$
$= 1.5$

$x_2(n)$:



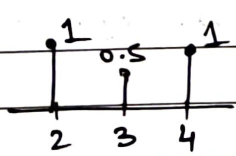
$\therefore \gamma_{x_1 x_2}(0) = (1\times1) + (1\times0.5) + (2\times1)$
$= 3.5$

$x_2(n-1)$:



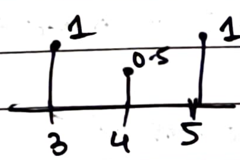
$\therefore \gamma_{x_1 x_2}(1) = (1\times1) + (2\times0.5) + (2\times1)$
$= 4$

$x_2(n-2)$:



$\therefore \gamma_{x_1 x_2}(2) = (2\times1) + (2\times0.5)$
$= 3$

$x_2(n-3)$:

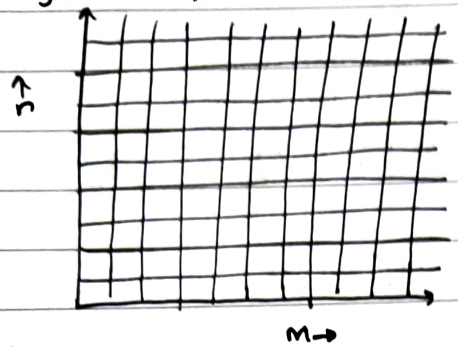


$\therefore \gamma_{x_1 x_2}(3) = (2\times1)$
$= 2$

* Image Processing ↓

Signals → time domain

images → spatial domain



And then each box represented as $f(m,n)$.

→ ↳ ~~Bitmap~~ Binary image (black-white image)

$0 \to$ black, $1 \to$ white

↳ Grayscale image (8 bits)

0 (black) – 255 (white) ⇒ Here bit-depth = 256

→ Total no. of bits required to represent image = [no. of rows × no. of columns × bit-depth]

⇒ ~~Point Processing~~ Spatial Domain Operations ↓

→ **Negative:**

Basically, ~~for each unit,~~

$g(m,n) = 255 - f(m,n)$ [if max. ~~ampli~~ intensity is 255]

→ **Image Addition:**

Basically, adding all corresponding units in 2 images together.

→ **Image Subtraction:**

Similar to addition, difference taken of all corresponding units (pixels) of 2 images.

→ [Note: Multiplication and Division are present too]

⇒ Point Processing ↓

(1) Brightness Modification :

$g(m,n) = f(m,n) \pm k$ (increase/decrease by constant k)

(2) Contrast Adjustment :

$g(m,n) = f(m,n) * k$ (multiply by constant k)

[$k>1$: increase contrast , $k<1$: decrease contrast]

(3) Thresholding:

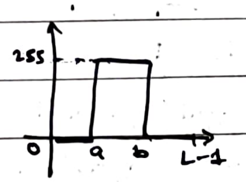
$s = 255$ if $r >= a$ (a= threshold value)

$s = 0$ if $r < a$

[used for converting grayscale image to binary image]

(4) Gray-level Slicing :

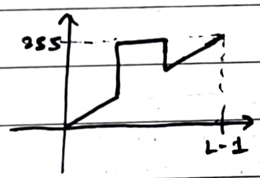
↳ without background →



$s = 255$ if $a <= r <= b$

$s = 0$ otherwise

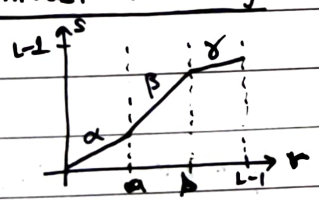
↳ with background →



$s = 255$ if $a <= r <= b$

$s = r$ otherwise

(5) Contrast stretching :



(6) Bit-Plane slicing :

MSB ↘ 0 0 0 ↙ LSB --- 0

1 0 0 --- 1

⋮ --- (2-6)

a a a --- 7

Separate binary images created from each bit positions of each pixel.

∴ For 8-bit image , 8 separate binary images created.

(7) Dynamic Range Compression : (Log transformation)

Log transformation $= c * \log(1 + |a|)$ [a = image]

(8) Power Law Transformation :

Display devices have non-linear ratio of intensity to voltage.

$s = c * r^{\gamma}$ (power is gamma)

⇒ Neighbourhood Processing (mask) ↴

$$\begin{array}{|c|c|c|} \hline w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ \hline w_{0,-1} & \boxed{w_{0,0}} & w_{0,1} \\ \hline w_{1,-1} & w_{1,0} & w_{1,1} \\ \hline \end{array} \leftarrow \text{Mask}$$

Each pixel on image is given a new value that is equal to sum of product of consecutive neighbours in the image and mask.

$$g(x,y) = \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} \left[w_{i,j} * f(x+i, y+j) \right]$$

→ Low Pass Averaging Filter :

$$\text{Mask} = \frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$ (sum of all mask coefficients = 1)

→ Weighted Averaging Filter :

In mask, more 'weight' given to cells closer to center cell (so corner cells/pixels given less 'weight' than center / side pixels)

Q. ~~10 10 10 10 10 10 10 10~~

$$\begin{bmatrix} 10 & 10 \\ 50 & 50 \end{bmatrix}$$ ← subset taken of original question.

Use low-pass averaging filter on this (3×3).

Ans. $\text{Mask} = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

∴ for 1st element : $\frac{1}{9} \times \begin{bmatrix} 10\times1 & 10\times1 & 10\times1 \\ 10\times1 & 10\times1 & 10\times1 \\ 50\times1 & 50\times1 & 50\times1 \end{bmatrix} = \frac{1}{9} \times 210 = 70/3$ //

for 2nd element : similarly we get $70/3$ //

for 3rd element : $1/9 \times [(10\times1)+(10\times1)+(10\times1)+(50\times1)+(50\times1)+(50\times1)+(50\times1)+(50\times1)+(50\times1)]$

$= 330/9 = 110/3$ //

Similarly for 4th element : $110/3$ //

∴ New matrix : $\begin{bmatrix} 70/3 & 70/3 \\ 110/3 & 110/3 \end{bmatrix}$ //

→ **Median Filter:**

Use mask, arrange all values in ascending order, replace center of mask with median value of the series.

→ **High Pass Filter:**

Eliminates low frequency regions and highlights high frequency regions.
(sum of all mask coefficients = 0)

Q. $\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 100 & 100 & 100 \end{bmatrix}$ ← subset of original question

Use high-pass filter on this (3x3)

Ans. Mask $= \frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

∴ For 1st element : $\frac{1}{9} \times [-10 -10 -10 -10 +80 -10 -10 -10 -10] = 0$
Similarly, for other elements in first row, we get 0
For 2nd row 1st element: $\frac{1}{9} \times [-10 -10 -10 -10 +80 -10 - 100 -100 -100] = -30$
Similarly for other elements in second row, we get -30
For 3rd row 1st element : $\frac{1}{9} \times [-10 -10 -10 -100 +800 -100 -100 -100 -100] = 30$
Similarly for other elements in third row, we get 30

∴ New matrix: $\begin{bmatrix} 0 & 0 & 0 \\ -30 & -30 & -30 \\ 30 & 30 & 30 \end{bmatrix}$ ← But we cannot have negative values in images.

∴ Final matrix: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 30 & 30 & 30 \end{bmatrix}$

→ **High Boost Filter:**

High Boost = $(A-1) * f(m,n)$ + High Pass

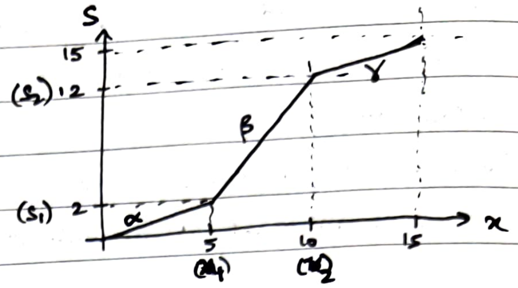
Mask : $\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9k-1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

→ **Zooming:**

Involves replicating pixels using neighbours.

Q. Perform contrast stretching on the image segment shown below according to the ~~transform~~ transform function given :

$$f(m,n) = \begin{bmatrix} 10 & 2 & 13 & 7 \\ 11 & 14 & 6 & 9 \\ 4 & 7 & 3 & 2 \\ 0 & 5 & 10 & 7 \end{bmatrix}$$



Ans. $\alpha = \frac{2}{5} = 0.4$

$\beta = (12-2)/(10-5) = 2$

$\gamma = (15-12)/(15-10) = 0.6$

Transformation function :

$s = \alpha \cdot x \quad , \quad 0 \le x \le x_1$

$s = \beta \cdot (x - x_1) + s_1 \quad , \quad x_1 < x \le x_2$

$s = \gamma \cdot (x - x_2) + s_2 \quad , \quad x_2 < x \le L-1$

$\therefore$ Going by pixel,

$(1,1) \rightarrow \beta(10-5)+2 = 10+2 = 12$

$(1,2) \rightarrow \alpha \cdot 2 = 0.4 \times 2 = 0.8 \approx 1$

$(1,3) \rightarrow \gamma(13-10)+12 = 1.8+12 = 13.8 \approx 14$

$(1,4) \rightarrow \beta(7-5)+2 = 4+2 = 6$

$(2,1) \rightarrow \gamma(11-10)+12 = 0.6+12 = 12.6 \approx 13$

$(2,2) \rightarrow \gamma(14-10)+12 = 2.4+12 = 14.4 \approx 14$

$(2,3) \rightarrow \beta(6-5)+2 = 2+2 = 4$

$(2,4) \rightarrow \beta(9-5)+2 = 8+2 = 10$

$(3,1) \rightarrow \alpha \cdot 4 = 0.4 \times 4 = 1.6 \approx 2$

$(3,2) \rightarrow$ From $(1,4) = 6$

$(3,3) \rightarrow \alpha \cdot 3 = 0.4 \times 3 = 1.2 \approx 1$

$(3,4) \rightarrow$ From $(1,2) = 1$

$(4,1) \rightarrow 0$

$(4,2) \rightarrow \alpha \cdot 5 = 0.4 \times 5 = 2$

$(4,3) \rightarrow$ From $(1,1) = 12$

$(4,4) \rightarrow$ From $(1,4) = 6$

Final image: $\begin{bmatrix} 12 & 1 & 14 & 6 \\ 13 & 14 & 4 & 10 \\ 2 & 6 & 1 & 1 \\ 0 & 2 & 12 & 6 \end{bmatrix}$

Q.

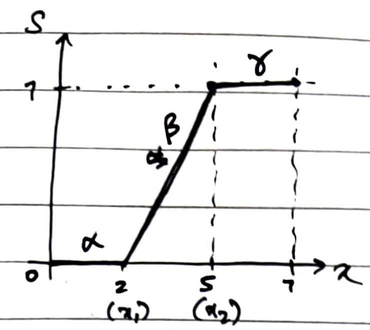


Image: $\begin{bmatrix} 3 & 3 & 4 & 4 & 5 & 6 \\ 4 & 5 & 4 & 1 & 2 & 4 \\ 3 & 4 & 3 & 2 & 1 & 0 \\ 2 & 3 & 3 & 4 & 4 & 4 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 0 & 7 & 0 & 7 & 0 & 0 \end{bmatrix}$

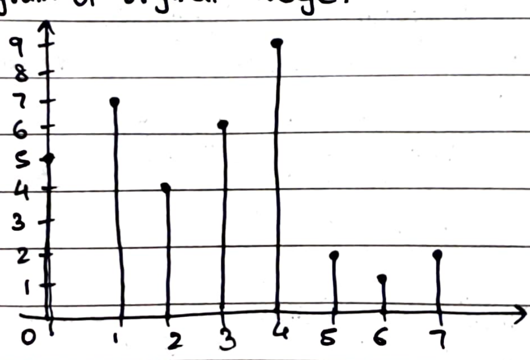
Ans. $\beta = 7/(5-2) = 7/3$, $\alpha = 0$, $\gamma = 7$

x	Condition Executed	S
0	$\alpha = 0 \quad \therefore 0 \times 0 = 0$	0
1	$\alpha = 0 \quad \therefore 0 \times 1 = 0$	0
2	$\alpha = 0 \quad \therefore 0 \times 2 = 0$	0
3	$\beta = 7/3 \quad \therefore 7/3(3-2) = 7/3 \approx 2$	2
4	$\beta = 7/3 \quad \therefore 7/3(4-2) = 14/3 \approx 5$	5
5	$\beta = 7/3 \quad \therefore 7/3(5-2) = 21/3 = 7$	7
6	$\gamma = 7 \quad \therefore 7(6-5)+7 = 14 \quad \therefore 7$	7
7	$\gamma = 7 \quad \therefore 7(7-5)+7 = 21 \quad \therefore 7$	7

$\therefore$ Using mapping from above table:

Final Image: $\begin{bmatrix} 2 & 2 & 5 & 5 & 7 & 7 \\ 5 & 7 & 5 & 0 & 0 & 5 \\ 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 7 & 0 & 0 \end{bmatrix}$ //

Histogram of original image:



(Absent one day →

* Image Transforms ↓

Spatial Domain → Frequency Domain

[Note: Real Term (X_R), Imaginary Term (X_I) } → { Magnitude $= \sqrt{X_R^2 + X_I^2}$, Phase $= \tan^{-1}(X_I/X_R)$]

Analog domain frequency: $-\infty$ to ∞
Digital domain frequency: 0 to 2π rad OR $-\pi$ to π rad OR $-\frac{1}{2}$ to $\frac{1}{2}$ cycles

→ Discrete Fourier Transform ↓ (DFT)

$\omega = \frac{2\pi}{N} k$, where $k = 0, \ldots, N-1$
N = No. of samples in DFT

$$\boxed{X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}k\right)n}}$$

→ usually calculated for no. of samples = (power of 2)

→ Inverse Discrete Fourier Transform ↓ (IDFT)

$$\boxed{x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\pi k}{N}\right)n}}$$

Q. $x(n) = [1, 0, 0, 1]$, $N=4$. Use DFT.

Ans. $X(k) = \sum_{n=0}^{3} x(n) e^{-j\left(\frac{2\pi k}{4}\right)n}$

$X(0) = x(0) e^{-j\left(\frac{2\pi 0}{4}\right)\cdot(0)} + x(1) e^{-j\left(\frac{2\pi 0}{4}\right)(1)} + x(2) e^{-j\left(\frac{2\pi}{4}\right)(0)(2)} + x(3) \cdot e^{-j\left(\frac{2\pi}{4}\right)(0)(3)}$

$\therefore X(0) = x(0) + x(1) + x(2) + x(3) = 1 + 0 + 0 + 1 = 2$

$X(1) = x(0) \cdot e^{-j\left(\frac{2\pi}{4}\right)(0)} + x(1) \cdot e^{-j\left(\frac{2\pi}{4}\right)(1)} + x(2) \cdot e^{-j\left(\frac{2\pi}{4}\right)(2)} + x(3) \cdot e^{-j\left(\frac{2\pi}{4}\right)(3)}$

$\therefore X(1) = x(0) + x(1) \cdot (-j) + x(2) \cdot (-1) + x(3) \cdot (j)$
$= 1 + 0 + 0 + j = 1 + j$

$X(2) = x(0) \cdot e^{-j(\pi)(0)} + x(1) \cdot e^{-j\pi(1)} + x(2) \cdot e^{-j(\pi)(2)} + x(3) \cdot e^{-j\pi(3)}$

$\therefore X(2) = 1 \cdot (1) + 0 \cdot (-1) + 0 \cdot (1) + 1 \cdot (-1) = 1 - 1 = 0$

$$X(3) = x(0)\cdot e^{-j\left(\frac{3\pi}{2}\right)(0)} + x(1)\cdot e^{-j\left(\frac{3\pi}{2}\right)(1)} + x(2)\cdot e^{-j\left(\frac{3\pi}{2}\right)(2)} + x(3)\cdot e^{-j\left(\frac{3\pi}{2}\right)(3)}$$

$$\therefore X(3) = 1\cdot(1) + 0\cdot(j) + 0\cdot(-1) + 1\cdot(-j) = 1 - j$$

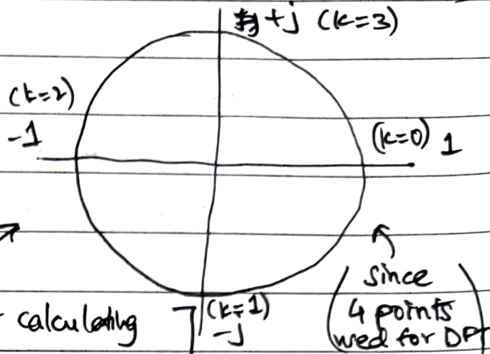
$$\therefore X(k) = [2, 1+j, 0, 1-j]$$

[Calculation was done using $e^{-j\omega n} = \cos(\omega n) - j\sin(\omega n)$]

kernel:

$$\begin{array}{c|cccc} k \downarrow \; n\rightarrow & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -j & -1 & +j \\ 2 & 1 & -1 & 1 & -1 \\ 3 & 1 & +j & -1 & -j \end{array}$$

(Basically value of $e^{-j\left(\frac{2\pi k}{N}\right)n}$ for each (k,n) pair respectively)



(since 4 points used for DFT)

[Can be used for calculating kernel, by judging how many sectors to go forward for each k and n value respectively]

~~So, X(k) = kernel x x(n)~~

~~[ ] = [ ] x [ ]~~ ~~4x1, 4x4, 4x1~~

So, $X(k) = \text{kernel} \times (x(n))$

$$\underbrace{[\quad]_{1\times4}}_{X(k)} = \underbrace{[\quad]_{1\times4}}_{x(n)} \times \underbrace{[\quad]_{4\times4}}_{\text{kernel}}$$

→ for 2D signals (images),

$f(m,n) \longrightarrow F(k,l)$

$$\boxed{F(k,l) = \text{kernel} \times f(m,n) \times (\text{kernel})^T} \leftarrow \text{DFT}$$

$$\boxed{f(m,n) = \frac{1}{N^2}\left[\text{kernel} \times F(k,l) \times (\text{kernel})^T\right]} \leftarrow \text{IDFT}$$

(next page) →

Q. $f(m,n) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ Obtain the 2D-DFT of the given image.

Ans. $F(k,l) = \text{kernel} \times f(m,n) \times (\text{kernel})^T$

We know ~~from~~ that,

4 point kernel $= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$ $\therefore (\text{kernel})^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$ (same)

$\cancel{\text{kernel} \times f(m,n) = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & & \\ & & & \end{bmatrix}}$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} //$$

Q. Calculate IDFT of previous answer.

Ans. $f(m,n) = \frac{1}{N^2}\left(\text{kernel} \times F(k,l) \times (\text{kernel})^T\right)$, $N=4$

$$\text{kernel} \times F(k,l) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix}$$

(continued)

$$[\text{kernel} \times F(k,l)] \times (\text{kernel})^T = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} = \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix}$$

$$\therefore \frac{1}{N^2} [\text{kernel} \times F(k,l) \times (\text{kernel})^T] = \frac{1}{16} \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Properties:

→ **Periodicity** -

$$x(n) = x(N+n) \xrightarrow{DFT} X(N+k)$$

→ **Linearity** -

$$\left.\begin{array}{l} x_1(n) \xrightarrow{DFT} X_1(k) \\ x_2(n) \xrightarrow{DFT} X_2(k) \end{array}\right\} \rightarrow a_1 x_1(n) + a_2 x_2(n) \xrightarrow{DFT} a_1 X_1(k) + a_2 X_2(k)$$

→ **Circular Convolution** -

$$x_1(n) \; N \; x_2(n) \xrightarrow{DFT} X_1(k) X_2(k)$$

[circular convolution where both $x_1(n)$ and $x_2(n)$ have period = N]

→ **Multiplication** -

$$x_1(n) x_2(n) \xrightarrow{DFT} X_1(k) \; N \; X_2(k)$$

→ **Time Reversal** -

$$(x(-n))_N \xrightarrow{DFT} (X(-k))_N$$

(periodicity) (periodicity)

→ **Circular Shift** -

$$(x(n-l))_N \xrightarrow{DFT} X(k) e^{-j\frac{2\pi k l}{N}}$$

→ Conjugate –

$$(x^*(n))_N \xrightarrow{DFT} X^*(N-k)$$

NOTE: Unitary transform if

$$A \times A^H = I \quad \text{(any constant multiple of identity mat.)}$$

where $A^H = A^{*T}$, I = identity matrix

A^* is complex conjugate of A, if $A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & +1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$, $A^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & +j & -1 & -j \\ 1 & -1 & +1 & -1 \\ 1 & -j & -1 & +j \end{bmatrix}$

Q. Find if DFT is a unitary transform.

Ans. kernel $= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$ ∴ kernel$^H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & +j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & +j \end{bmatrix}$

∴ kernel × kernel$^H = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} = 4 \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ //

∴ DFT kernel satisfies Unitary transform condition. //