6532142421

Pupipat Singkhorn

<<< Only Problem 2 and 4 will be graded >>>

In [ ]:
```
import matplotlib.pyplot as plt
plt.style.use('seaborn-v0_8-whitegrid')
import numpy as np
import IPython.display as ipd
%matplotlib inline
import os
from scipy import signal,fftpack
from skimage.io import imread
import cv2
```

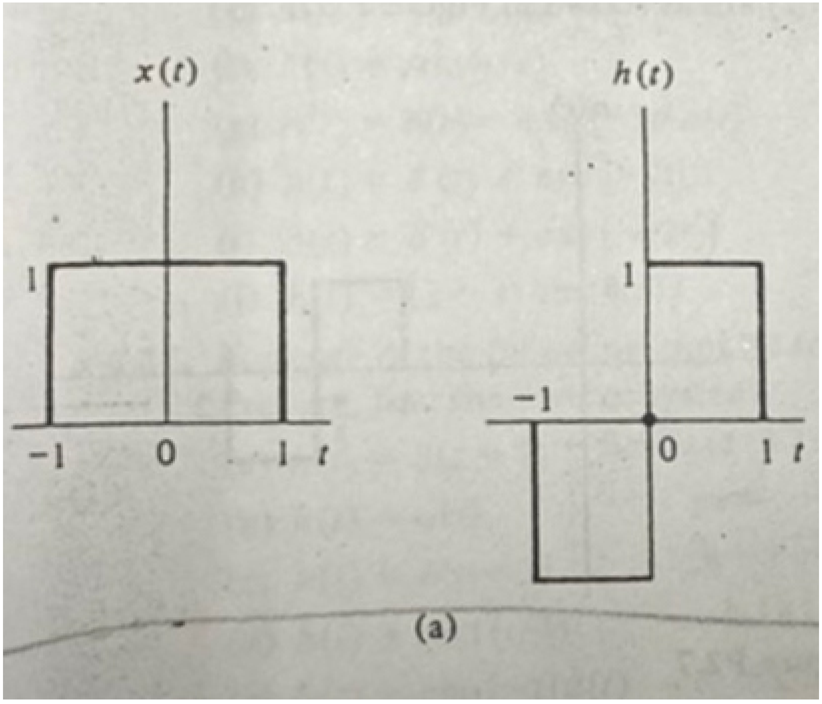
Problem 1

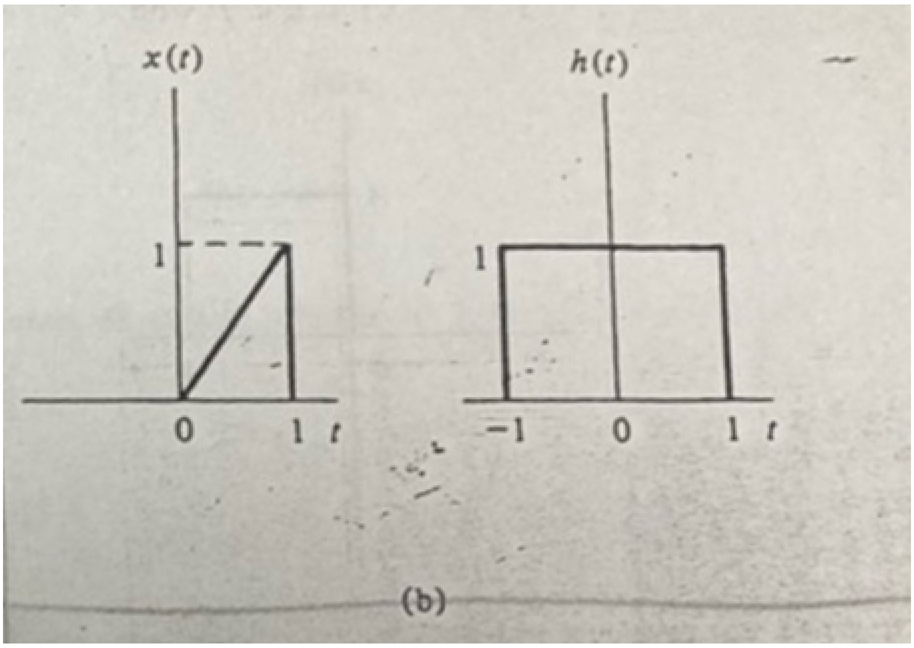
Evaluate the convolution of the following signals

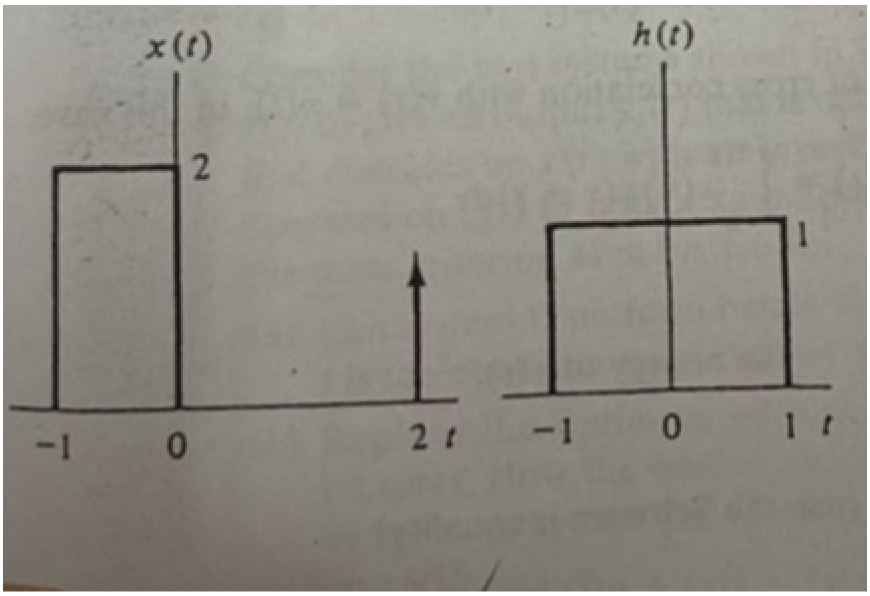
1. $\mathrm{rect}\left(\frac{t-a}{a}\right) * \delta(t-b)$
2. $\mathrm{rect}\left(\frac{t}{a}\right) * \mathrm{rect}\left(\frac{t}{a}\right)$
3. $t[u(t)-u(t-1)] * u(t)$

Problem 2

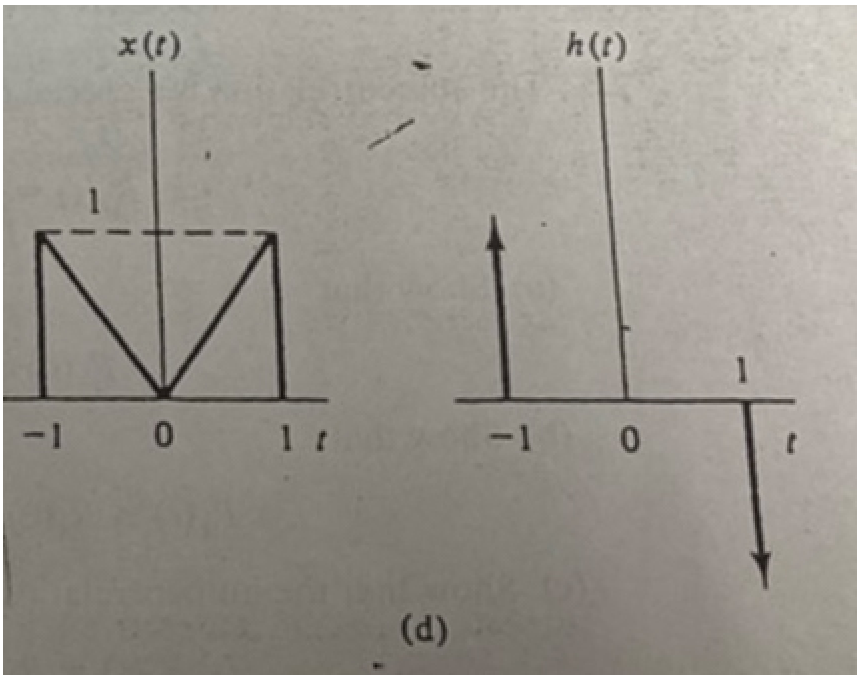
Determine the convolution $y(t) = h(t) * x(t)$ using Graphical Interpretation of the pairs of the signals shown

1. 

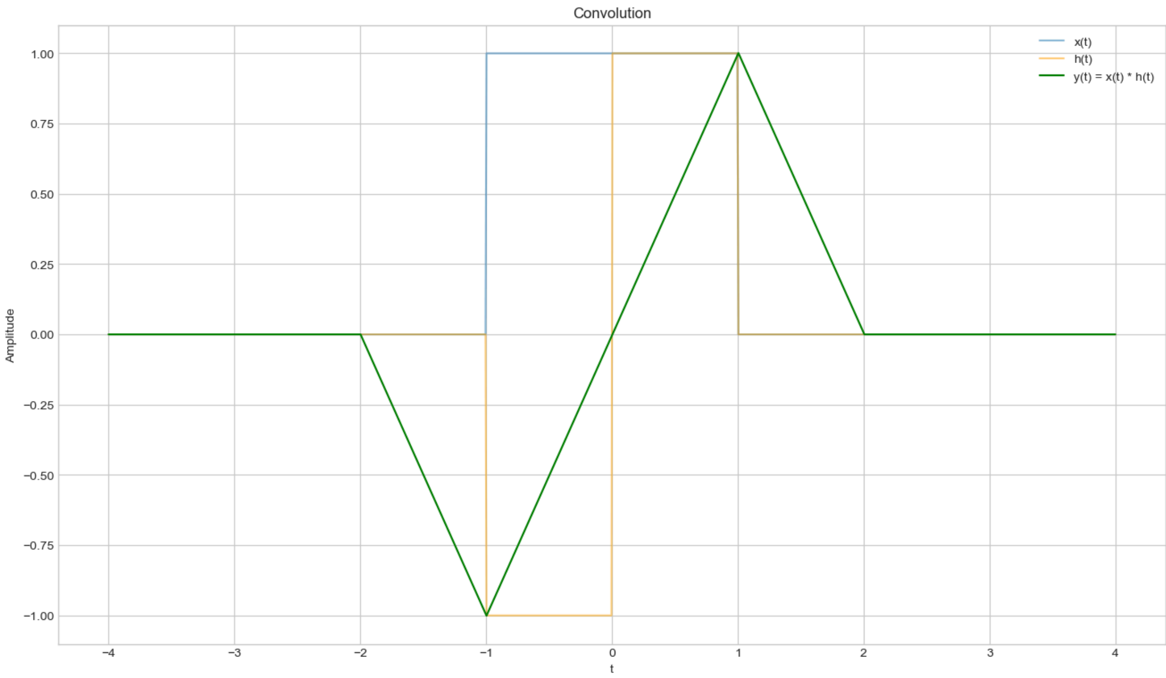
2. 

3. 

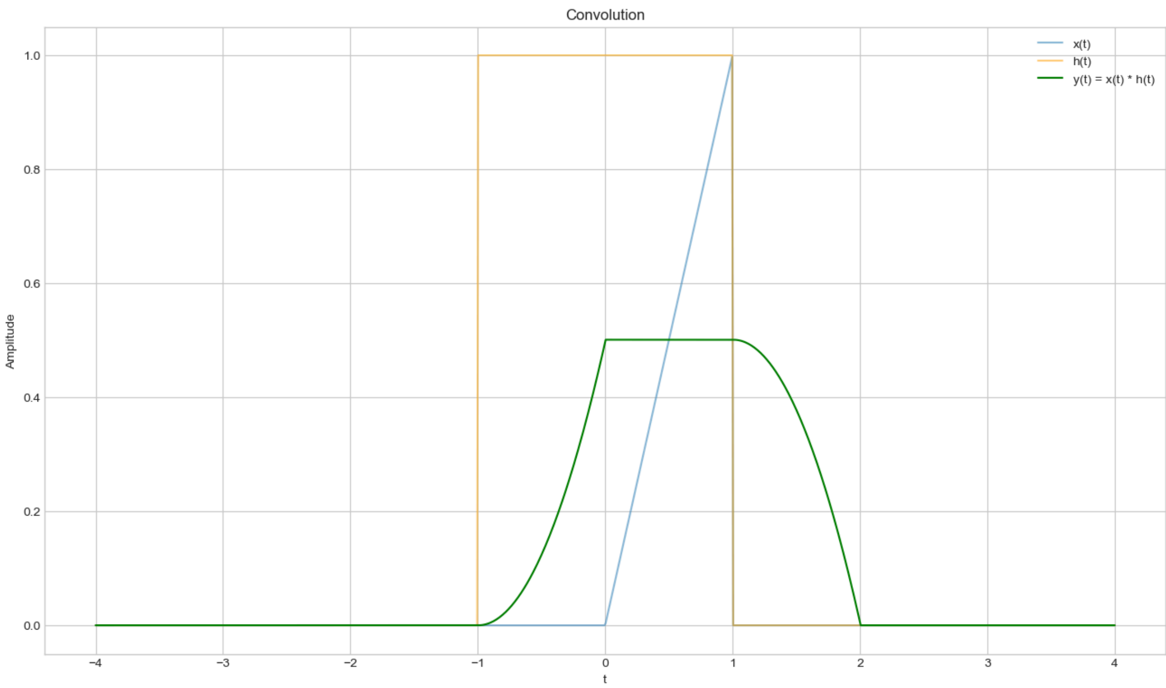
[optional]

4. 


```
In [ ]: # 1.

t = np.linspace(-4, 4, 1000)

def x(t):
    return np.where((t >= -1) & (t <= 1), 1, 0)

def h(t):
    return np.where((t >= 0) & (t <= 1), 1, 0) - np.where((t >= -1) & (t

y = np.convolve(x(t), h(t), mode='same') * (t[1] - t[0]) # Normalize by t

def plot(t, x, h, y):
    plt.figure(figsize=(16, 9))
    plt.plot(t, x(t), label='x(t)', alpha=0.5)
    plt.plot(t, h(t), label='h(t)', color='orange', alpha=0.5)
    plt.plot(t, y, label='y(t) = x(t) * h(t)', color='green')
    plt.title('Convolution')
    plt.xlabel('t')
    plt.ylabel('Amplitude')
    plt.grid(True)
    plt.legend()
    plt.show()

plot(t, x, h, y)
```

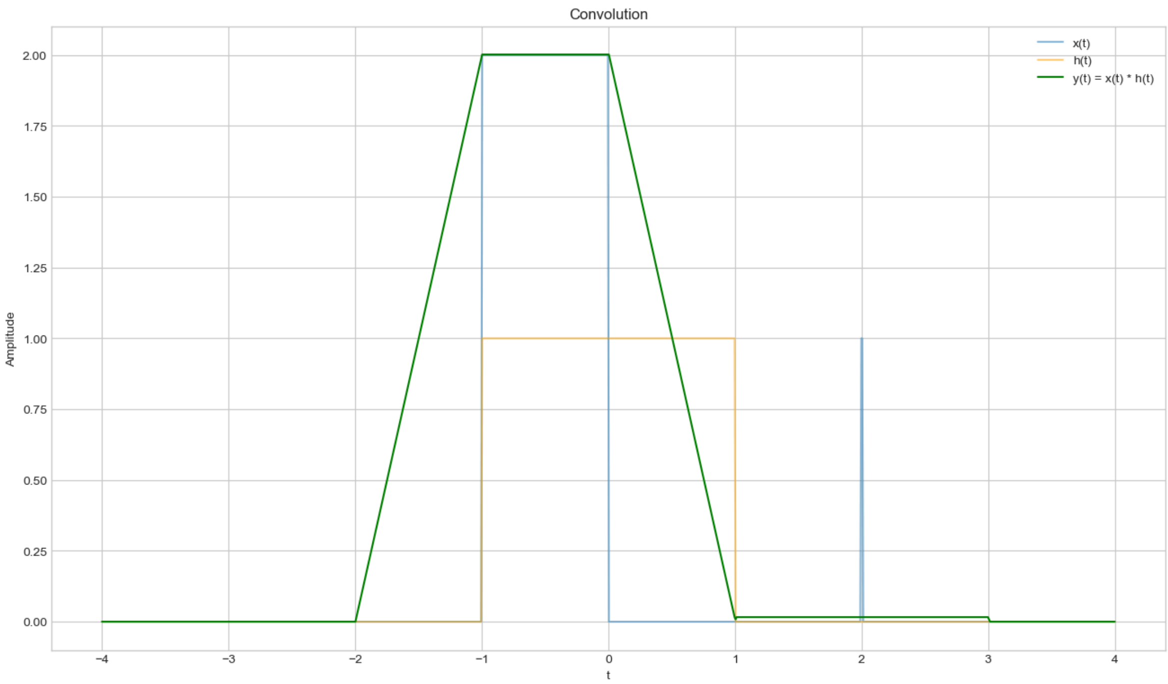


In [ ]:
```
# 2.

t = np.linspace(-4, 4, 1000)

def x(t):
    return np.where((t >= 0) & (t <= 1), t, 0)

def h(t):
    return np.where((t >= -1) & (t <= 1), 1, 0)

y = np.convolve(x(t), h(t), mode='same') * (t[1] - t[0]) # Normalize by t

plot(t, x, h, y)
```



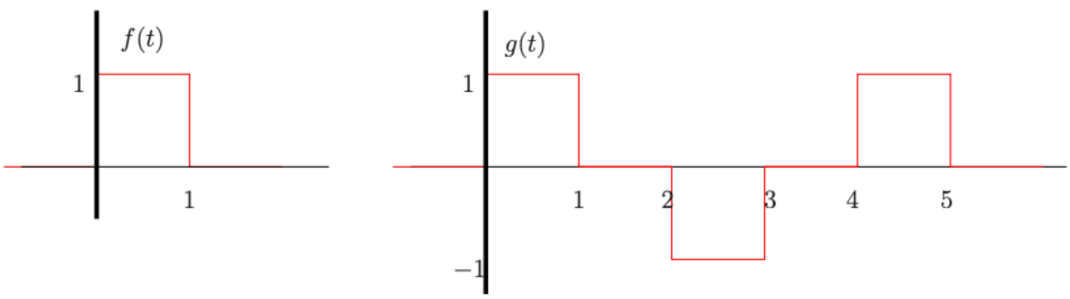
In [ ]:
```
# 3.

t = np.linspace(-4, 4, 1000)
```

```
def x(t):
    rect = np.where((t >= -1) & (t <= 0), 2, 0)

    idx = np.argsort(np.abs(t - 2))[:2]
    delta = np.zeros_like(t)
    delta[idx] = 1 # Unit impulse

    return rect + delta

def h(t):
    return np.where((t >= -1) & (t <= 1), 1, 0)

y = np.convolve(x(t), h(t), mode='same') * (t[1] - t[0]) # Normalize by t

plot(t, x, h, y)
```



Problem 3

Let $f(t)$ and $g(t)$ be given as follows:



1. sketch the function : $x(t)$ = $f(t) * g(t)$
2. show that if $a(t) = b(t) * c(t)$,then $(Mb(t)) * c(t) = Ma(t)$, for any real number M (hint: use the convolution integral formula)

Problem 4

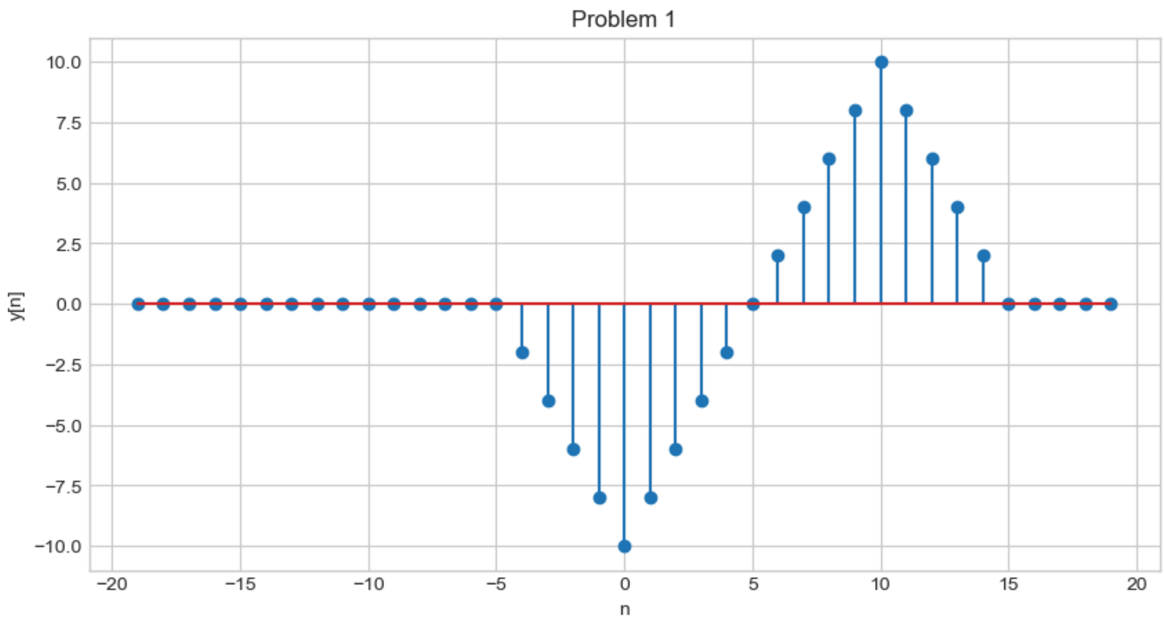
Find the convolution $y[n] = h[n] * x[n]$ of the following signals:

In [ ]:
```
# Unit step function
def u(n):
    return np.where(n >= 0, 1, 0)
```

1.

$$x[n] = \begin{cases} -1, -5 \leq n \leq -1 \\ 1, 0 \leq n \leq 4 \end{cases} \quad h[n] = 2u[n]$$

In [ ]:
```
n1 = np.arange(-10, 10)
x1 = np.where((n1 >= -5) & (n1 <= -1), -1, 0) + np.where((n1 >= 0) & (n1
h1 = 2 * u(n1)

y1 = signal.convolve(x1, h1, mode='full')

plt.figure(figsize=(10, 5))
plt.stem(np.arange(len(y1)) - len(h1) + 1, y1)
plt.title('Problem 1')
plt.xlabel('n')
plt.ylabel('y[n]')
plt.grid(True)
plt.show()
```

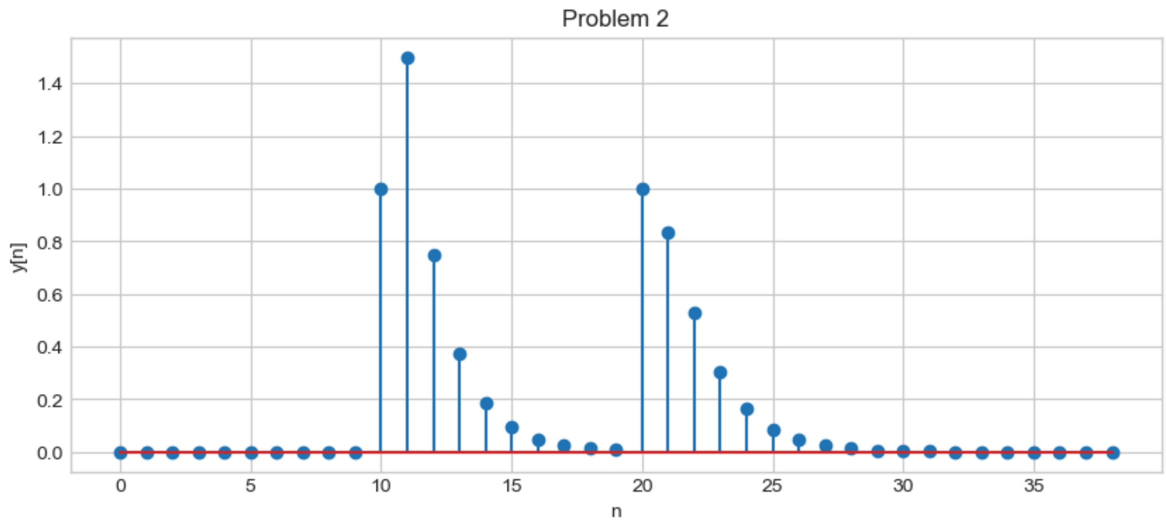


2.

$$x[n] = \left(\frac{1}{2}\right)^n u[n],\ h[n] = \delta[n] + \delta[n-1] + \left(\frac{1}{3}\right)^n u[n]$$

In [ ]:
```
n2 = np.arange(-10, 10)
x2 = (1/2)**n2 * u(n2)
h2 = signal.unit_impulse(20, 0) + signal.unit_impulse(20, 1) + (1/3)**n2

y2 = signal.convolve(x2, h2, mode='full')
```

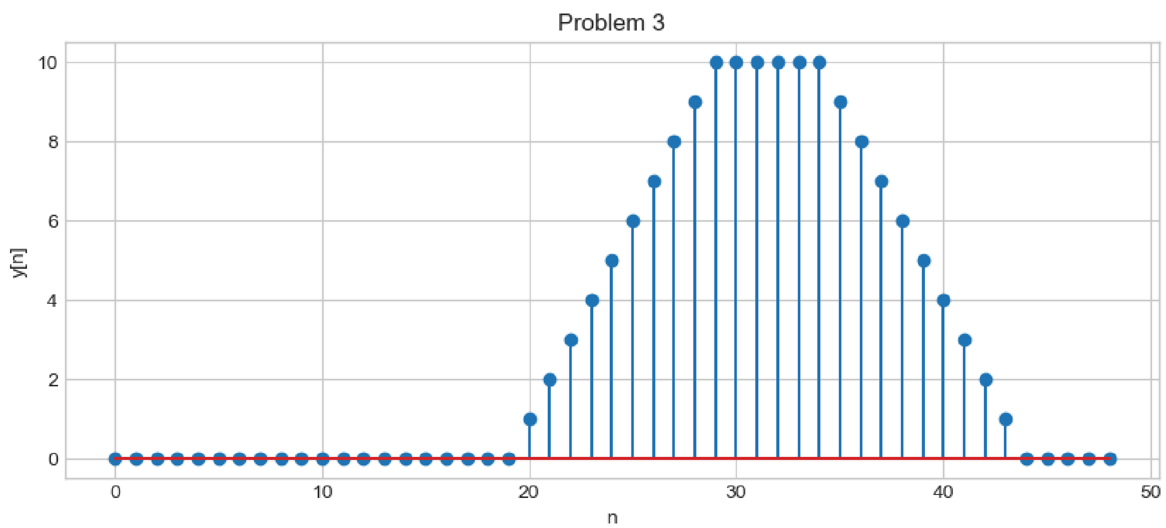
```python
plt.figure(figsize=(10, 4))
plt.stem(np.arange(len(y2)), y2)
plt.title('Problem 2')
plt.xlabel('n')
plt.ylabel('y[n]')
plt.grid(True)
plt.show()
```



3.

$x[n] = u[n]$, $h[n] = 1\ ; 0 \leq n \leq 9$

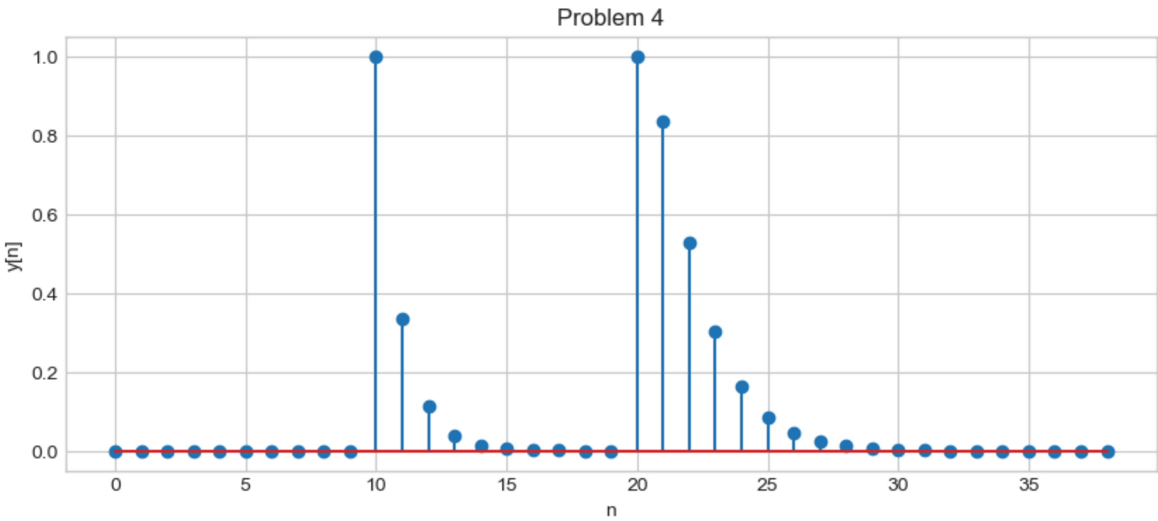
In [ ]:
```python
n3 = np.arange(-10, 15)
x3 = u(n3)
h3 = np.where((n3 >= 0) & (n3 <= 9), 1, 0)

y3 = signal.convolve(x3, h3, mode='full')

plt.figure(figsize=(10, 4))
plt.stem(np.arange(len(y3)), y3)
plt.title('Problem 3')
plt.xlabel('n')
plt.ylabel('y[n]')
plt.grid(True)
plt.show()
```



4.

$$x[n] = \left(\frac{1}{3}\right)^n u[n],\, h[n] = \delta[n] + \left(\frac{1}{2}\right)^n u[n]$$

In [ ]:
```
n4 = np.arange(-10, 10)
x4 = (1/3)**n4 * u(n4)
h4 = signal.unit_impulse(20, 0) + (1/2)**n4 * u(n4)

y4 = signal.convolve(x4, h4, mode='full')

plt.figure(figsize=(10, 4))
plt.stem(np.arange(len(y4)), y4)
plt.title('Problem 4')
plt.xlabel('n')
plt.ylabel('y[n]')
plt.grid(True)
plt.show()
```



Problem 5

Find the convolution $y[n] = h[n] * x[n]$ of the following signals

1.

$$x[n] = \left\{1, -\frac{1}{2}, \frac{1}{4}, -\frac{1}{8}, \frac{1}{16}\right\},\, h[n] = \{1, -1, 1, -1\}$$

1.

$$x[n] = \{1, 2, 3, 0, -1, \},\, h[n] = \{2, -1, 3, 1, -2\}$$

1.

$$x[n] = \left\{3, \frac{1}{2}, -\frac{1}{4}, 1, 4\right\},\, h[n] = \left\{2, -1, \frac{1}{2}, -\frac{1}{2}\right\}$$

1.

$$x[n] = \left\{-1, \frac{1}{2}, \frac{3}{4}, -\frac{1}{5}, 1\right\},\, h[n] = \{1, 1, 1, 1, 1\}$$

Problem 6

Problem 6.1 : Convolution - 1D

The following code creates a gaussian pulse and its self convolutions. Study and apply the convolution between signal e and another signal e with noise (e_noise) and write the report to analyze the results.

In [ ]:
```
t = np.linspace(-1, 1, 2 * 100, endpoint=False)
i, q, e = signal.gausspulse(t, fc=5, retquad=True, retenv=True)
plt.plot(t, e, '--',label = 'orignal signal')
plt.legend(loc='upper right')
plt.show()


conv_e = np.convolve(e,e,'same')
plt.plot(t, e, '--',label = 'orignal signal')
plt.plot(t, conv_e, '--',label = 'self conv signal')
plt.legend(loc='upper right')
plt.show()

e_noise = e + np.random.randn(len(e))*2.5
conv_e_noise = np.convolve(e,e_noise,'same')

# TODO : Apply the convolution between signal e and another signal e with
```

Problem 6.2

From the self convolution below, when increasing the number of self convolution (now is 8), what is noticeable from the final shape resulted from the convolution?

(HINT 01: Central limit theorem)

(HINT 02: What is Probability Density Function (PDF) of z if $z = x + y$?)

In [ ]:
```
from scipy.stats import uniform

x = np.linspace(-5,5, 1000)
plt.plot(x, uniform.pdf(x),
       'r-', lw=5, alpha=0.6, label='uniform pdf')

plt.show()


x = np.linspace(-15,15, 10000)
pdf_1 = uniform.pdf(x)
pdf_2 = uniform.pdf(x)

for i in range(8):
    pdf_2 = np.convolve(pdf_2,pdf_1, 'same')

pdf_2 = pdf_2/np.max(pdf_2)
plt.plot(x, pdf_2,'r-', lw=5, alpha=0.6, label='conv uniform')
```

Problem 7

2D (image) signal convolution:

The following code show the 2D signal (image f(x,y)) and a kernel (diag_line). Study the convolution of the kernel and the image. Apply with "circuits.png" image and analyze the results.

TODO : Apply diag_line to the "circuits.png image" and analyse the results

In [ ]:

```
!wget https://drive.google.com/uc?id=1hQ8uKocLTjaKmrJm-O4BfZxvLLM1Cfa- -O
!wget https://drive.google.com/uc?id=1WoISJ6-FECt-gt6Ovjlfmz89oGH812GM -O
```

In [ ]:

```
image_path = 'hamtaro0.jpg'

diag_line = np.array([[ 2, -1, -1],
                      [-1, 2, -1],
                      [-1, -1, 2]])

ham = cv2.imread(image_path,0)
plt.figure(figsize=(10,10))
plt.imshow(ham, cmap='gray')
plt.show()
grad = signal.convolve2d(ham,diag_line,boundary='symm',mode='same')
plt.figure(figsize=(10,10))
plt.imshow(grad, cmap='gray')
plt.show()

# TODO : Apply diag_line to the "circuits.png image" and analyse the resu
```

Problem 8

Are the following systems linear or time invariant?

1. $x(t)$ -> **System(a)** -> $7x(t-1)$
2. $x(t)$ -> **System(b)** -> $cos(2x(t))$
3. $x(t)$ -> **System(c)** -> t
4. $x(t)$ -> **System(d)** -> $x(t)+t$

In [ ]: