Lab 2a

Name : Abhinav Pandey

Roll No : AM.EN.U4AIE21088

1) Perform the convolution of 1D digital signal with np.convolve function.

In [ ]:
```
import numpy as np

sequence1 = np.array([1, 2, 3])
sequence2 = np.array([4, 5, 6])

result = np.convolve(sequence1, sequence2, mode='full')

print(result)
```

```
[ 4 13 28 27 18]
```

2) Try the above example with 'same' and 'valid' modes. Write down the difference.

In [ ]:
```
import numpy as np

a  = np.array([3,7])

v = np.array([1,2,5,7])

print("First array: ", a)
print("Second array: ", v)

conv_full = np.convolve(a,v, mode='full')
conv_same = np.convolve(a,v, mode='same')
conv_valid = np.convolve(a,v, mode='valid')
print("Convolution Full: ", conv_full)
print("Convolution Same: ", conv_same)
print("Convolution Valid: ", conv_valid)
```

```
First array:  [3 7]
Second array:  [1 2 5 7]
Convolution Full:  [ 3 13 29 56 49]
Convolution Same:  [ 3 13 29 56]
Convolution Valid:  [13 29 56]
```

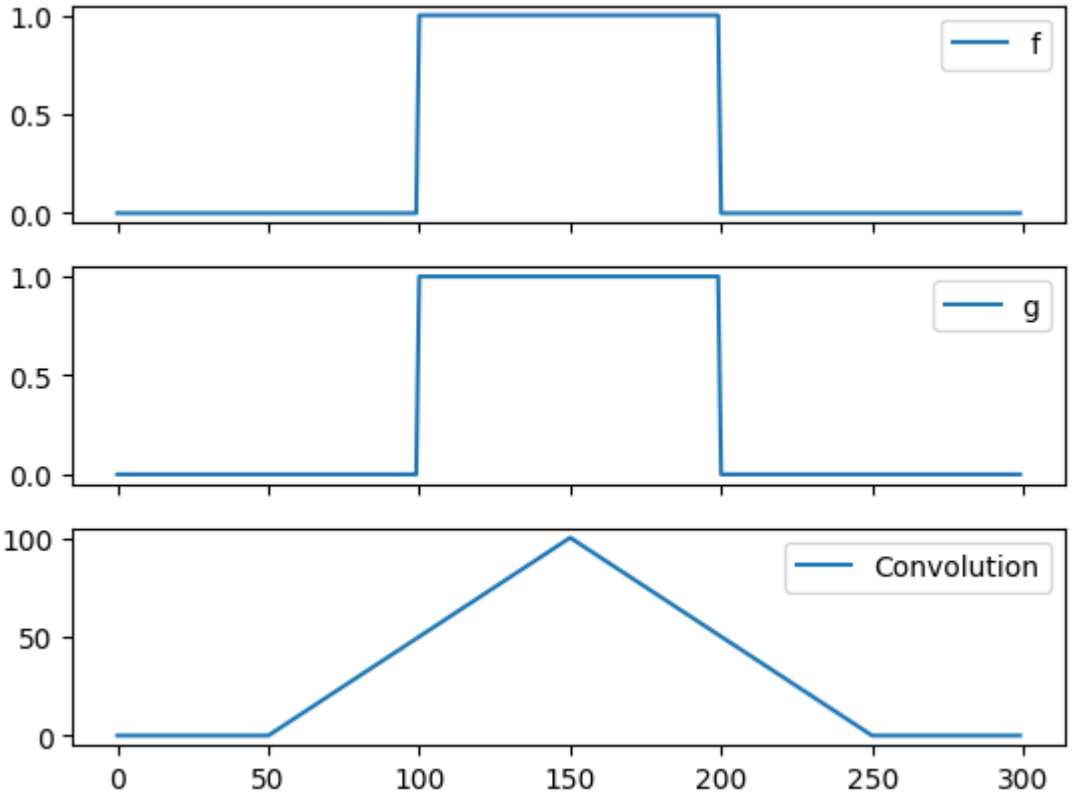
The difference are as follows:

- `full` computes the convolution over all possible overlaps, resulting in a larger output.
- `same` pads the input arrays to ensure complete overlap, resulting in an output of the same size as the largest input array.

- `valid` only considers positions where the input arrays fully overlap, resulting in a smaller output size.

3) Perform 1d convolution of two square signal with same width.

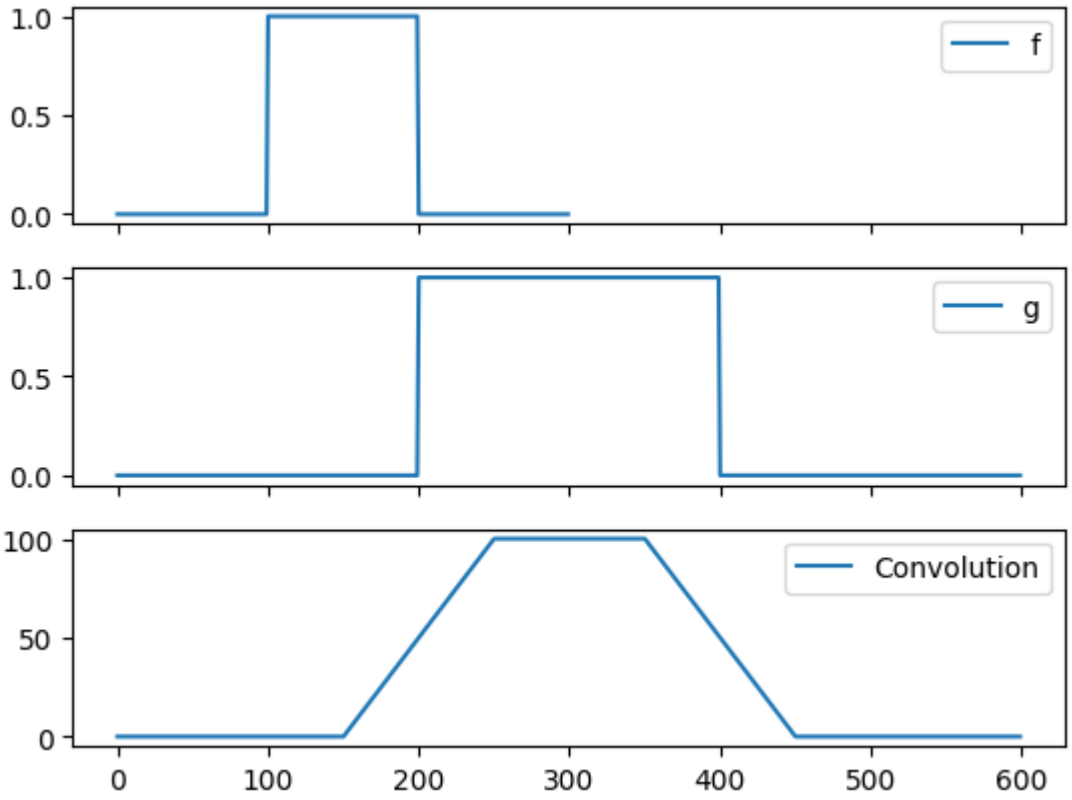
In [ ]:
```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
sig1 = np.repeat([0., 1., 0.], 100)
sig2 = np.repeat([0., 1., 0.], 100)
filtered = np.convolve(sig1, sig2, mode='same')
fig, ax = plt.subplots(3,1, sharex=True)
ax[0].plot(sig1, label='f')
ax[1].plot(sig2, label='g')
ax[2].plot(filtered, label = 'Convolution')
for axx in ax:
 axx.legend()
plt.savefig('convolve-same.png',bbox_inches='tight', dpi=300)
```



4) Perform 1D convolution of two square signal where one signal width is twice as that of the first one. What is the shape of the resultant signal?

In [ ]:
```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
sig1 = np.repeat([0., 1., 0.], 100)
sig2 = np.repeat([0., 1., 0.], 200)
filtered = np.convolve(sig1, sig2, mode='same')
fig, ax = plt.subplots(3,1, sharex=True)
ax[0].plot(sig1, label='f')
ax[1].plot(sig2, label='g')
ax[2].plot(filtered, label = 'Convolution')
```

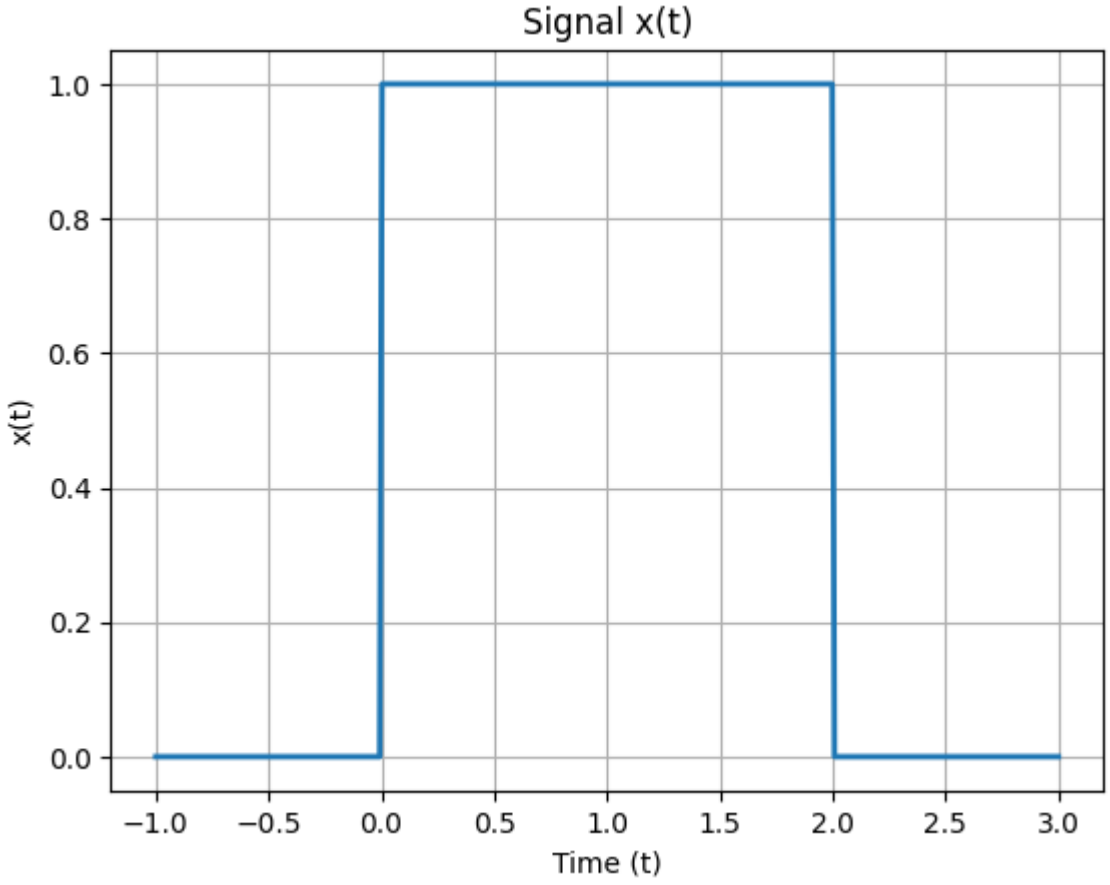
```
for axx in ax:
 axx.legend()
plt.savefig('convolve-200.png',bbox_inches='tight', dpi=300)
```



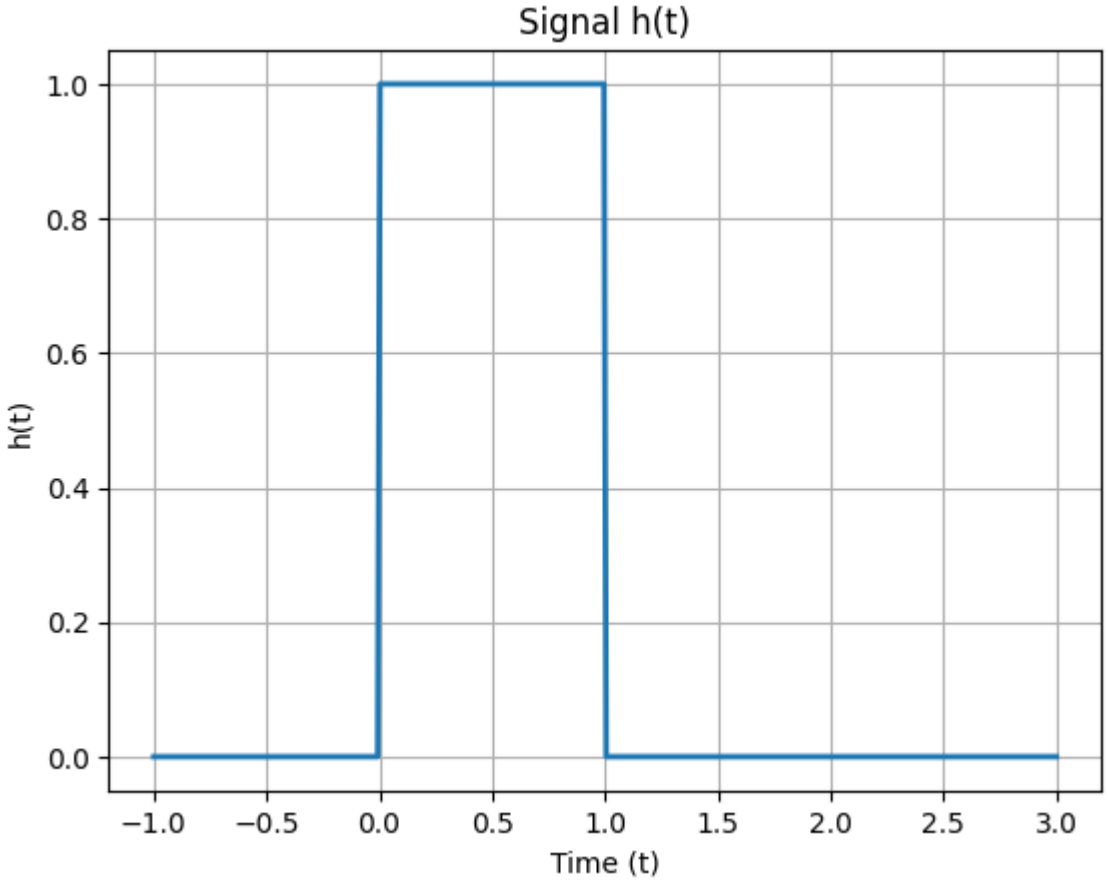
Class assignments

Question A

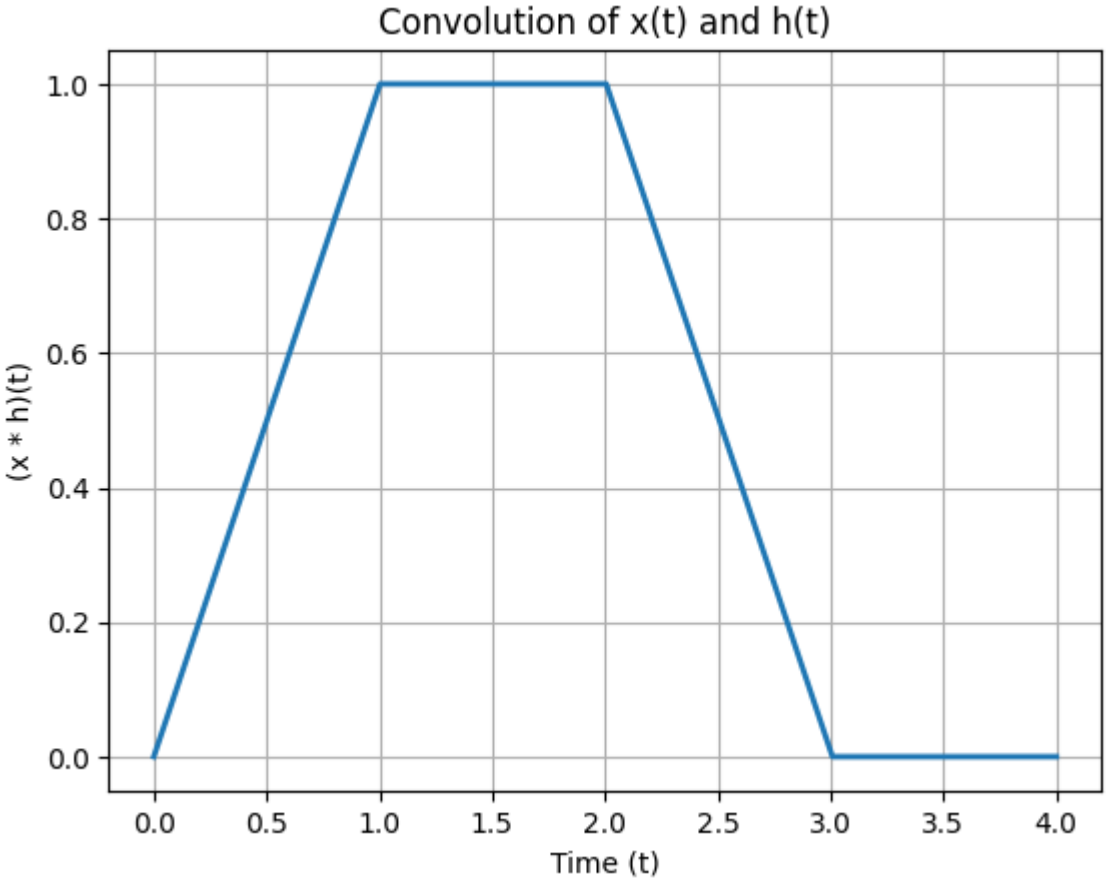
In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-1, 3, 400)
x = np.piecewise(t, [t < 0, (t >= 0) & (t <= 2), t > 2], [0, 1, 0])

plt.plot(t, x, lw=2)
plt.title('Signal x(t)')
plt.xlabel('Time (t)')
plt.ylabel('x(t)')
plt.grid(True)
plt.show()
```



In [ ]:
```
h = np.piecewise(t, [t < 0, (t >= 0) & (t <= 1), t > 1], [0, 1, 0])

plt.plot(t, h, lw=2)
plt.title('Signal h(t)')
plt.xlabel('Time (t)')
plt.ylabel('h(t)')
plt.grid(True)
plt.show()
```

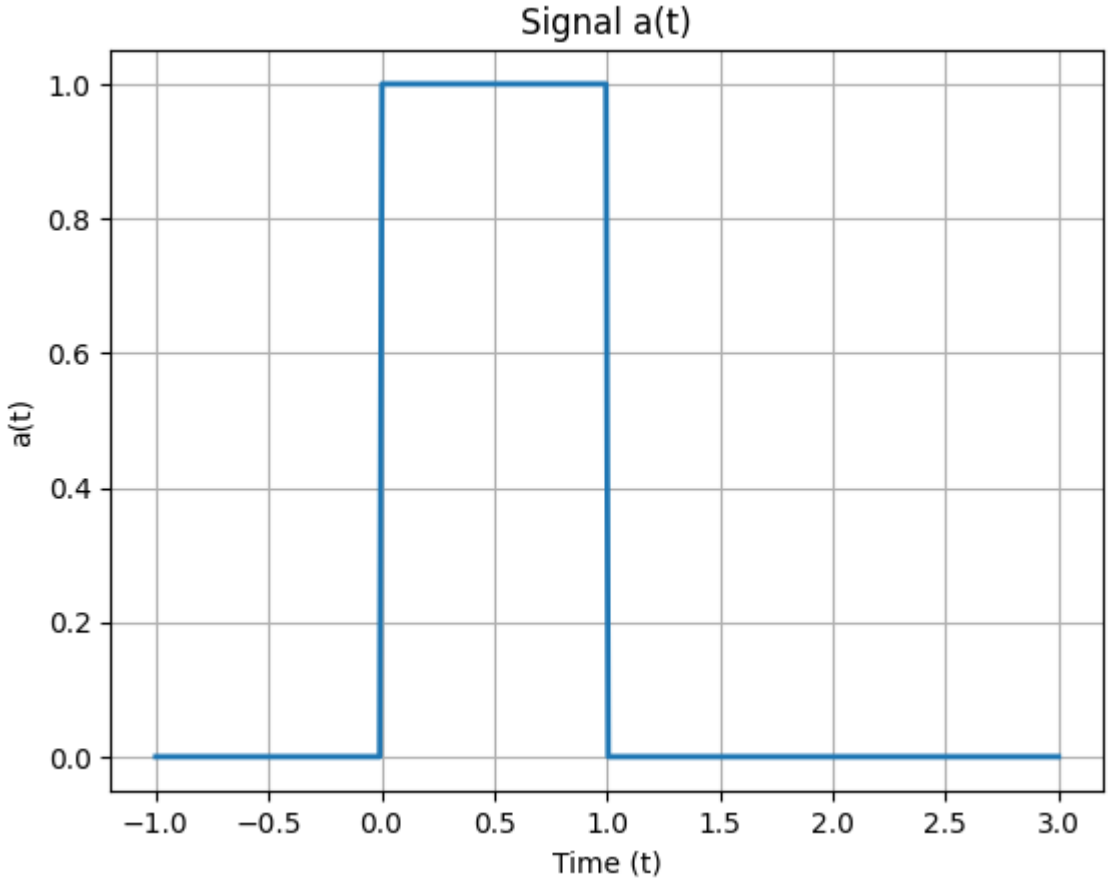


In [ ]:
```
t = np.linspace(0, 4, 400)
convolution = np.convolve(x, h, mode='same')/sum(h)

plt.plot(t, convolution, lw=2)
plt.title('Convolution of x(t) and h(t)')
plt.xlabel('Time (t)')
plt.ylabel('(x * h)(t)')
plt.grid(True)
plt.show()
```

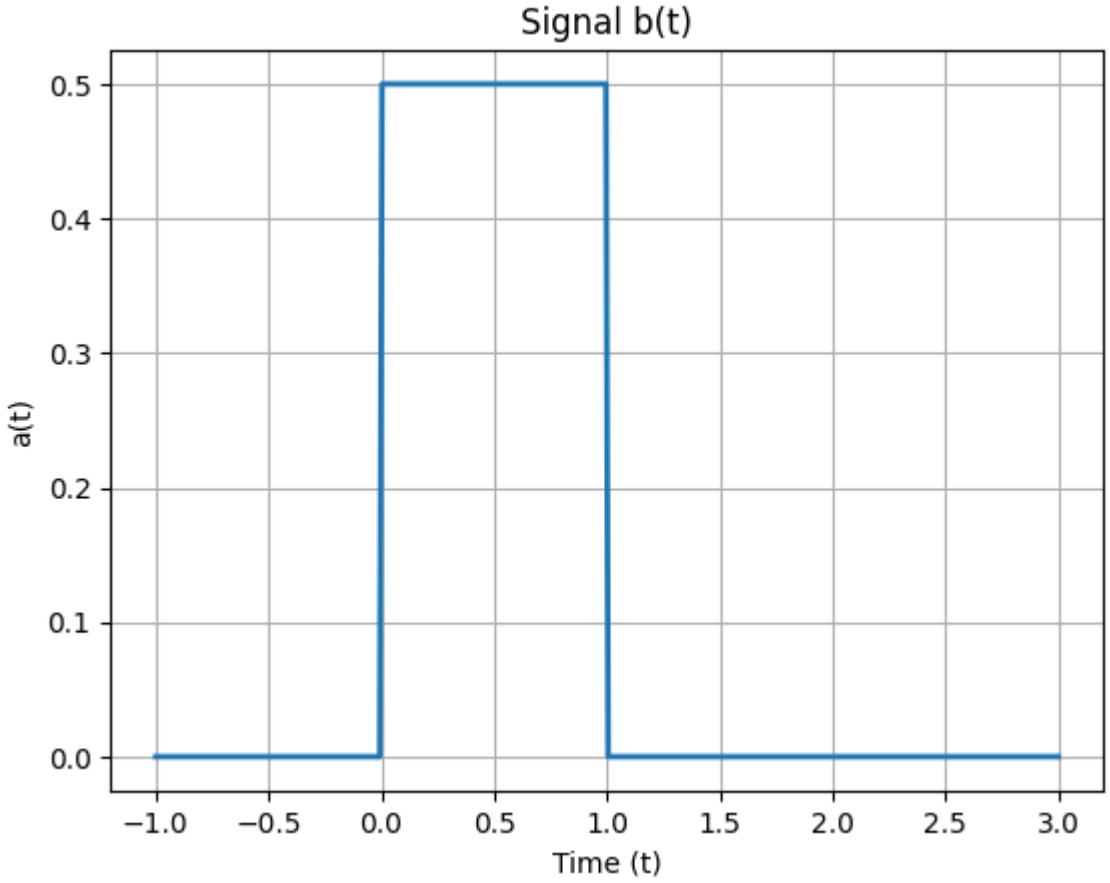


Question B

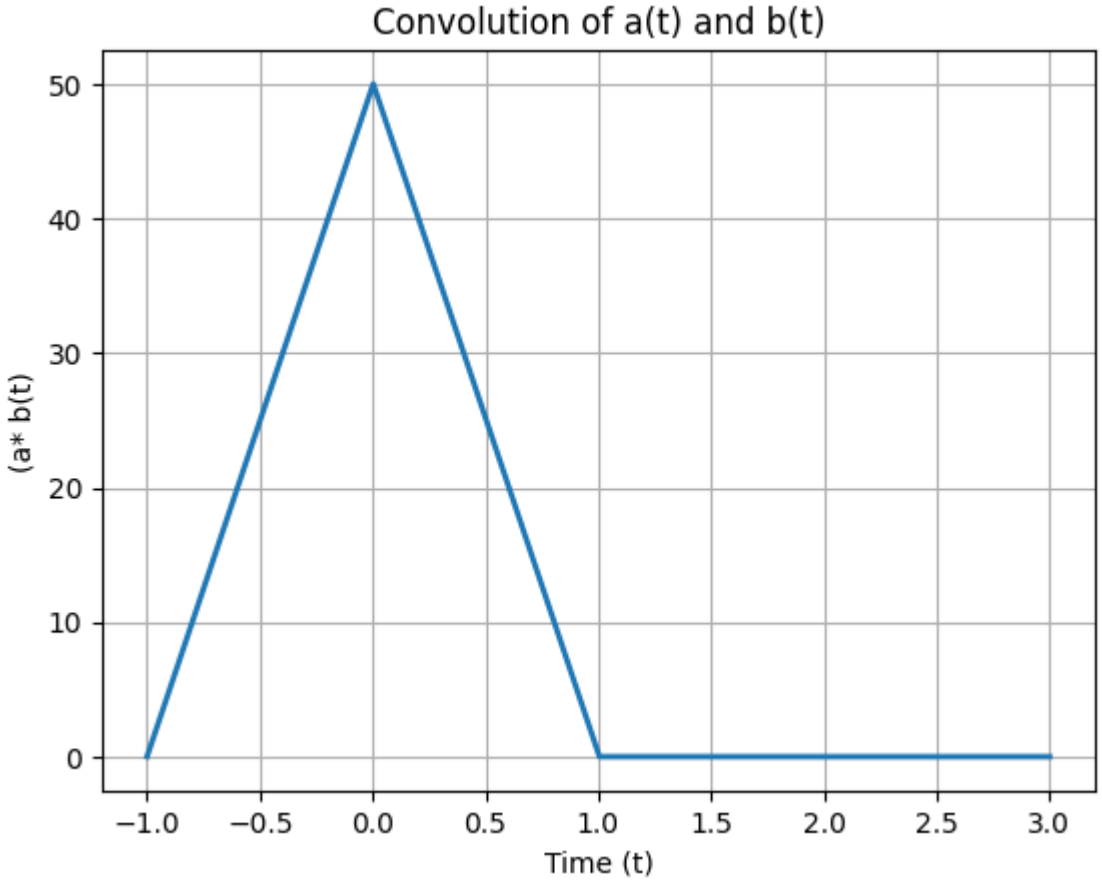
In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-1, 3, 400)
a = np.piecewise(t, [t < 0, (t >= 0) & (t <= 1), t > 1], [0, 1, 0])

plt.plot(t, a, lw=2)
plt.title('Signal a(t)')
plt.xlabel('Time (t)')
plt.ylabel('a(t)')
plt.grid(True)
plt.show()
```



In [ ]:
```
b = np.piecewise(t, [t < 0, (t >= 0) & (t <= 1), t > 1], [0, 0.5, 0])

plt.plot(t, b, lw=2)
plt.title('Signal b(t)')
plt.xlabel('Time (t)')
plt.ylabel('a(t)')
plt.grid(True)
plt.show()
```



In [ ]:
```
convolution = np.convolve(a, b, mode='same')

t = np.linspace(-1, 3, 400)
plt.plot(t, convolution, lw=2)
plt.title('Convolution of a(t) and b(t)')
plt.xlabel('Time (t)')
plt.ylabel('(a* b(t)')
plt.grid(True)
plt.show()
```


Convolution of a(t) and b(t)
(a* b(t)
Time (t)
50
40
30
20
10
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0