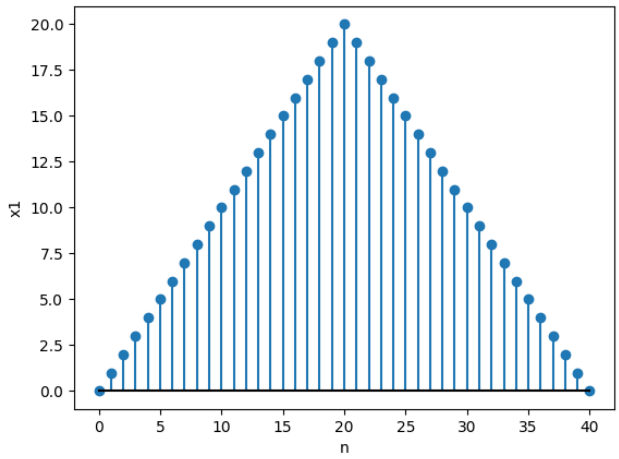
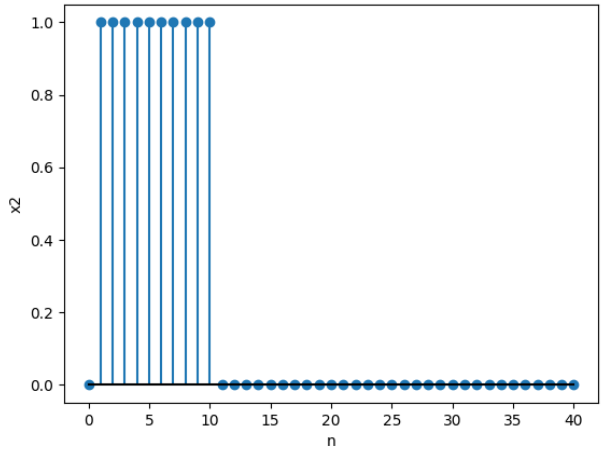
Part A

$$x_1[n] = \begin{cases} n, & 1 \le n \le 20 \\ 40 - n, & 21 \le n \le 39 \\ 0, & \text{otherwise} \end{cases}$$

$$x_2[n] = u[n-1] - u[n-11]$$

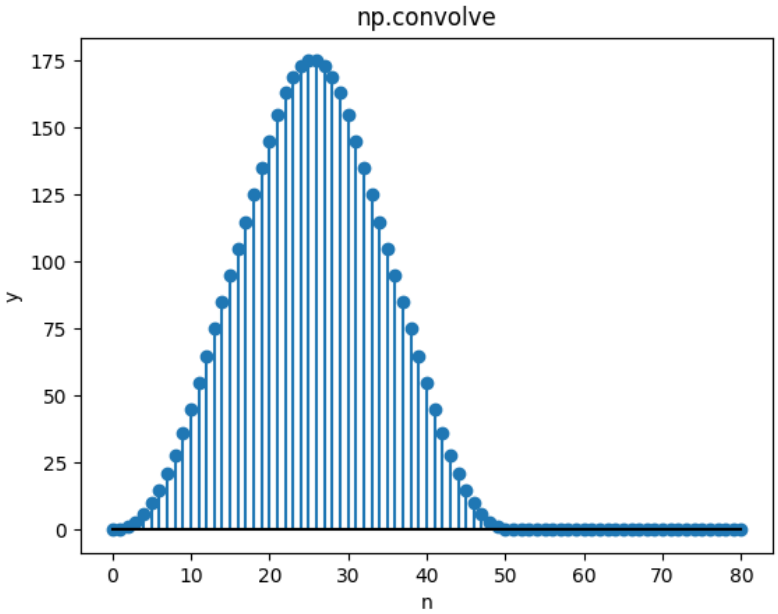
Plot x1[n] and x2[n] vs n (n=0 ~ n=40)





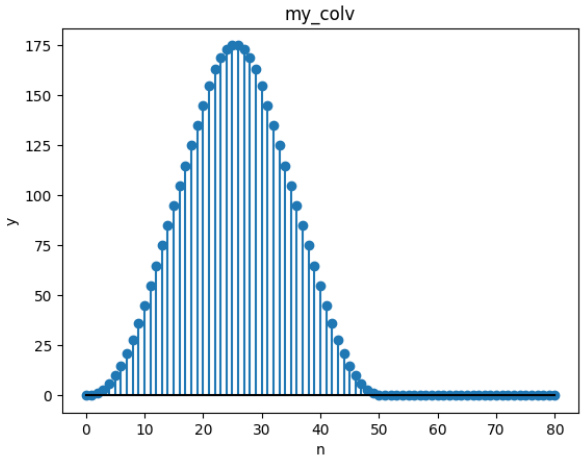
Part B

Use numpy.convolve to get the convolution sum of x1[n] and x2[n] (n = 0 ~ (41+41-1)-1)



Part C

Use the python function I implement to get the convolution sum of x1[n] and x2[n]



Print(y1.all() == y2.all()) is to check whether the elements in y1 and y2 are all the same.

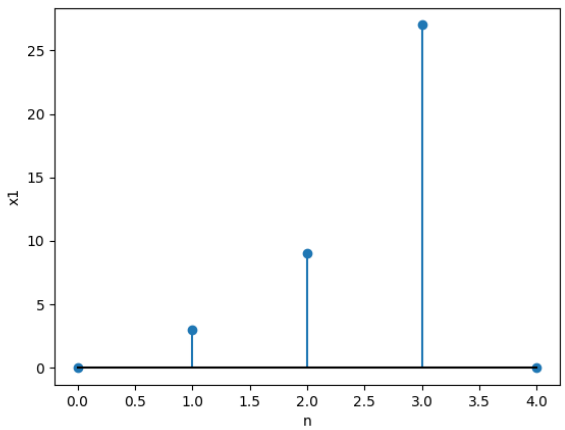
```
### Part b
y1 = np.convolve(x1, x2, mode='full')
n = np.arange(0, len(y1))
draw(n, y1, 'n', 'y', 'np.convolve')

### Part c
y2 = my_colv(x1, x2)
draw(n, y2, 'n', 'y', 'my_colv')
print(y1.all() == y2.all())
```

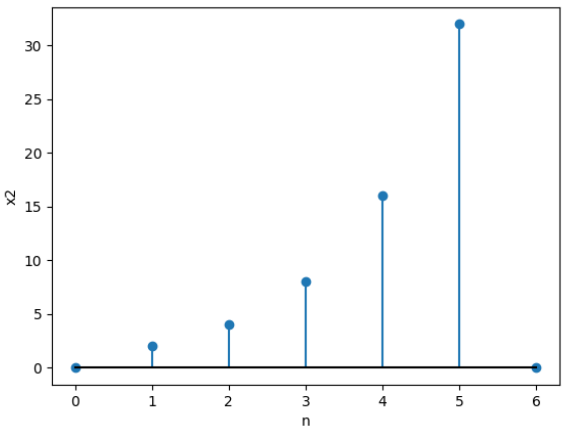
Part D

$$x_1[n] = \begin{cases} 3^n u[n], & 1 \le n \le 3 \\ 0, & \text{otherwise} \end{cases}$$

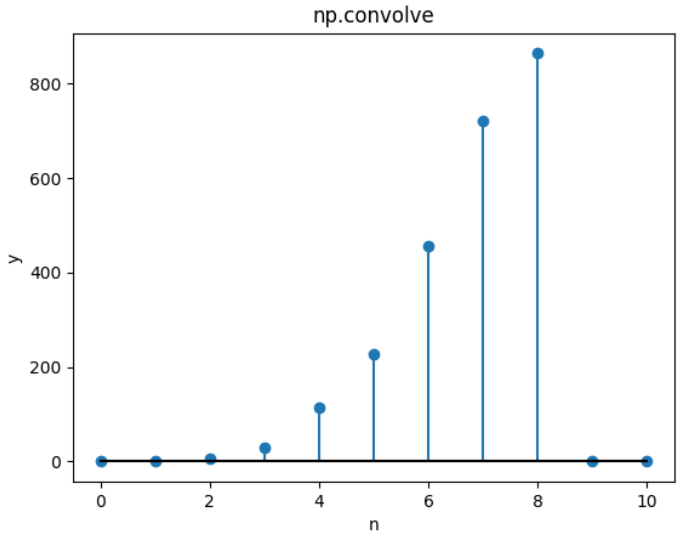
$$x_2[n] = \begin{cases} 2^n u[n], & 1 \le n \le 5 \\ 0, & \text{otherwise} \end{cases}$$

Plot x1[n] and x2[n] vs n (n = 0~4 and 0~6)

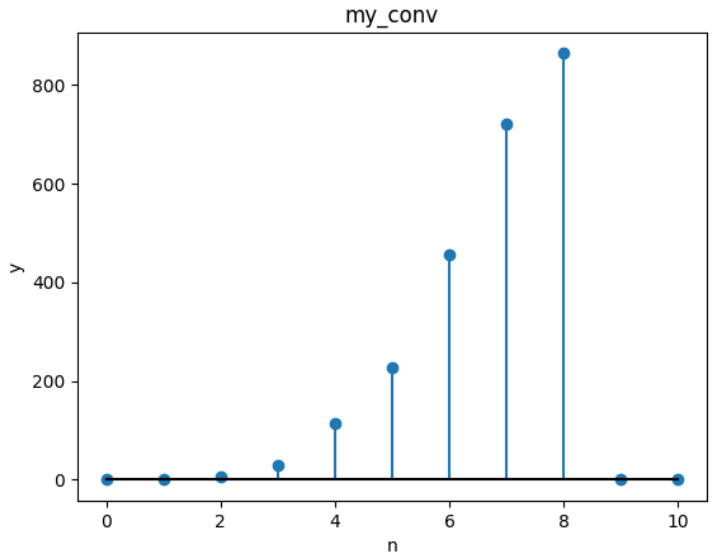




Use numpy.convolve to get the convolution sum of x1[n] and x2[n] (n = 0 ~ (5+7-1)-1)



Use the python function I implement to get the convolution sum of x1[n] and x2[n]



Print(y1.all() == y2.all()) is to check whether the elements in y1 and y2 are all the same.

```
y1 = np.convolve(x1, x2, 'full')
n = np.arange(0, len(y1))
draw(n, y1, 'n', 'y', 'np.convolve')

y2 = my_colv(x1, x2)
draw(n, y2, 'n', 'y', 'my_conv')

print(y1.all() == y2.all())
```