	Marwari University **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Simulate Linear Convolution and Circular Convolution on Discrete Time Signals.	
Experiment No: 02	**Date: 12-08-2024**	**Enrollment No: 92200133030**

Aim: Simulate Linear Convolution and Circular Convolution on Discrete Time Signals.

Theory:-

- Linear convolution and circular convolution are mathematical operations that combine two signals to obtain a third signal. They are widely used in various applications, such as signal processing, image processing, and audio processing.
- Linear convolution calculates the sum of element-wise products of two signals, considering the full range of valid indices. It is typically used for finite-length signals and can produce an output signal that is longer than the input signals.
- Circular convolution, on the other hand, calculates the sum of element-wise products of two signals, considering a periodic extension of the input signals. It is commonly used for periodic or infinite-length signals and produces an output signal with the same length as the input signals.

Programm:-

```
import matplotlib.pyplot as plt
import numpy as np

def linear_convolution(signal1, signal2):
    # Compute the linear convolution
    linear_conv = np.convolve(signal1, signal2, mode='full')
    return linear_conv

def circular_convolution(signal1, signal2):
    # Compute the circular convolution
    fft_length = max(len(signal1), len(signal2))

    fft_signal1 = np.fft.fft(signal1, fft_length)
    fft_signal2 = np.fft.fft(signal2, fft_length)
    circular_conv = np.fft.ifft(fft_signal1 * fft_signal2)

    return circular_conv

# Define the discrete-time signals
signal1 = np.array([1, 2, 3, 4, 5])
signal2 = np.array([2, 4, 6, 8, 10])

# Compute the linear convolution
linear_conv = linear_convolution(signal1, signal2)


# Compute the circular convolution
circular_conv = circular_convolution(signal1, signal2)
```

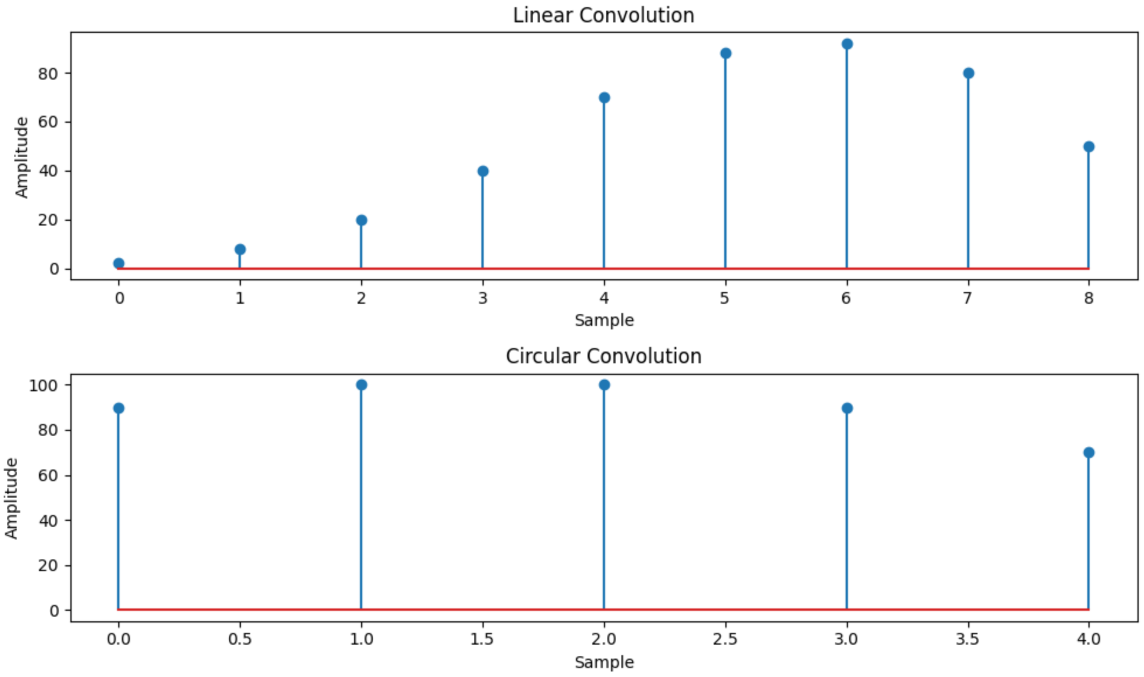
```
# Plot the linear and circular convolution results
plt.figure(figsize=(10, 6))

plt.subplot(2, 1, 1)
plt.stem(linear_conv)
plt.title('Linear Convolution')
plt.xlabel('Sample')
plt.ylabel('Amplitude')

plt.subplot(2, 1, 2)
# Use the real part of the circular convolution
plt.stem(circular_conv.real)
plt.title('Circular Convolution')
plt.xlabel('Sample')
plt.ylabel('Amplitude')

plt.tight_layout()
plt.savefig("./Convolution.png")
plt.show()
```

Output :-


Linear Convolution
Amplitude
Sample
Circular Convolution
Amplitude
Sample