In [1]:
```
# 1. Importing the Necessary Libraries

import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image,ImageFilter
from scipy. ndimage.filters import convolve
from google.colab.patches import cv2_imshow
```

```
<ipython-input-1-9bbfb984027e>:7: DeprecationWarning: Please import `convolve` fr
om the `scipy.ndimage` namespace; the `scipy.ndimage.filters` namespace is deprec
ated and will be removed in SciPy 2.0.0.
  from scipy. ndimage.filters import convolve
```

In [2]:
```
# 2. Importing the Image

image_path = '/content/Image.jpg'
img = cv2.imread(image_path ,  cv2.IMREAD_GRAYSCALE)
image_float = np.array(img, dtype=np.float64)
cv2_imshow(img)
```

Now Implementing Differnet types of Kernals

1. Identity Kernel

In [3]:
```
print("Kernel: Identity Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[0, 0, 0],
                   [0, 1, 0],
                   [0, 0, 0]])
```

```
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Identity Kernel
```

2. Box Blur (Average Filter)

In [4]:
```
print("Kernel: Box Blur Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = (1/9) * np.array([[1, 1, 1],
                           [1, 1, 1],
                           [1, 1, 1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Box Blur Kernel
```

3. Gaussian Blur

In [5]:
```
print("Kernel: Gaussian Blur Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = (1/16) * np.array([[1, 2, 1],
                            [2, 4, 2],
                            [1, 2, 1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Gaussian Blur Kernel
```

4. Sobel X (Horizontal Edge Detection)

In [6]:
```
print("Kernel: Sobel X Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[-1, 0, 1],
                   [-2, 0, 2],
                   [-1, 0, 1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Sobel X Kernel
```

5. Sobel Y (Vertical Edge Detection)

In [7]:
```
print("Kernel: Sobel Y Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[-1, -2, -1],
                   [0, 0, 0],
                   [1, 2, 1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Sobel Y Kernel
```

6. Laplacian (Edge Detection)

In [8]:
```
print("Kernel: Laplacian Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[0, -1, 0],
                   [-1, 4, -1],
                   [0, -1, 0]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Laplacian Kernel
```

7. Sharpen

In [9]:
```
print("Kernel: Sharpen Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[0, -1, 0],
                   [-1, 5, -1],
                   [0, -1, 0]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Sharpen Kernel
```

8. High-Pass Filter

In [10]:
```
print("Kernel: High-Pass Filter Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[-1, -1, -1],
                   [-1, 8, -1],
                   [-1, -1, -1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: High-Pass Filter Kernel
```

9. Motion Blur

In [11]:
```
print("Kernel: Motion Blur Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = (1/9) * np.array([[1, 0, 0],
                           [0, 1, 0],
                           [0, 0, 1]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Motion Blur Kernel
```

10. Emboss

In [12]:
```
print("Kernel: Emboss Kernel")
image_float = np.array(img, dtype=np.float64)
kernal = np.array([[-2, -1, 0],
                   [-1, 1, 1],
                   [0, 1, 2]])
im2 = convolve(image_float, kernal)
image_bounded = np.array(np.clip(im2, 0, 255), dtype=np.uint8)
img_new = cv2.hconcat([img, image_bounded])
cv2_imshow(img_new)
```

```
Kernel: Emboss Kernel
```