```python
import numpy as np
import cv2
from scipy.fftpack import fftshift, ifftshift, fft2, ifft2
from google.colab.patches import cv2_imshow

def butterworth_highpass_filter(shape, cutoff, order):
    P, Q = shape
    H = np.zeros((P, Q), dtype=np.float32)
    for u in range(P):
        for v in range(Q):
            D_uv = np.sqrt((u - P/2)**2 + (v - Q/2)**2)
            H[u, v] = 1 / (1 + (cutoff / D_uv)**(2 * order))
    return H

def apply_filter(image, filter):
    dft = fftshift(fft2(image))
    dft_filtered = dft * filter
    image_filtered = np.abs(ifft2(ifftshift(dft_filtered)))
    return image_filtered

# Load image in grayscale
image = cv2.imread('/content/Ganeshji (1).webp', cv2.IMREAD_GRAYSCALE)

# Define Butterworth high-pass filter parameters
cutoff = 30  # Cutoff frequency
order = 2    # Filter order

# Create Butterworth high-pass filter
filter = butterworth_highpass_filter(image.shape, cutoff, order)

# Apply filter to image
filtered_image = apply_filter(image, filter)

# Normalize and convert to uint8
filtered_image = cv2.normalize(filtered_image, None, 0, 255, cv2.NORM_MINMAX)
filtered_image = np.uint8(filtered_image)

# Save or display the result
cv2_imshow(image)
cv2_imshow(filtered_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
<ipython-input-1-394962bd95b3>:12: RuntimeWarning: divide by zero encountered in scalar divide
  H[u, v] = 1 / (1 + (cutoff / D_uv)**(2 * order))
```



```
<ipython-input-1-394962bd95b3>:12: RuntimeWarning: divide by zero encountered in scalar divide
  H[u, v] = 1 / (1 + (cutoff / D_uv)**(2 * order))
```