This code implements image watermarking embedding and extraction based on the Arnold transform and the discrete Hartley transform (DHT). The main steps are as follows:   
  
**Initialization:** Set block size, load images, and initialize parameters. Load the main image and the watermark image, calculate the total number of bits of the watermark, and prepare the position for embedding the watermark.   
  
**Watermarking Process:** Apply Lorenz chaotic mapping encryption (Hundun function) to the watermark image and extract the watermark bit data for each color layer (gainlevelwatermark).   
  
**Watermark embedding:** The color layers of the main image are processed block by block. For each block, a DHT transform is performed, and the watermark information is embedded in the DHT coefficients of the block using a Hadamard matrix for both forward and inverse transformations. The watermark information is embedded by modifying the DHT coefficients, so that the watermark is closely combined with the image content.   
  
**PSNR and SSIM calculation:** The PSNR (peak signal-to-noise ratio) and SSIM (structural similarity) values of the watermarked image are calculated to evaluate the image quality after the watermark is embedded.   
  
**Watermark Extraction:** Extract the watermark from an image with embedded watermark by first performing a IDHT transform on each color layer, dequantizing, and recovering the watermark bits. The integrity of the watermark is verified by comparing the extracted watermark information with the original watermark.   
  
**Performance Evaluation:** The PSNR and SSIM values are calculated to evaluate the quality loss of the image and the extraction effect of the watermark. The original image and the watermarked image are displayed, and the relevant indicators are output.   
  
In summary, the code implements effective embedding and extraction of watermarks in images while maintaining high image quality through complex transformations and quantization.