First, I use structure defines the header structure of a bitmap file, including file identifiers, file size, image width, and height, etc. And then I set a function print_header function: This function is used to print the header information of an image file. Later, I set write_image_file function: This function is used to write processed image data to a file, including writing header information and pixel data to the file.

Main function includes processes such as image reading, quarter reduction, and merging transformation. The implementation of image shrinking involves dividing the width and height of the original image by 2, and then sampling each pixel. This means that for each pixel in the original image, its value is selected as the pixel value at the corresponding position in the reduced image according to a certain rule (such as every other pixel). The final size of the reduced image is half of the original image.

The implementation of image merging involves the following steps:

Determine the size of the merged image, which is usually the sum of the size of the original image and three times the size of the reduced image (assuming merging to the upper-right quadrant, similar for other quadrants).

Create a new image data structure and allocate enough memory space.

Traverse the original image and the reduced image, and merge their pixels into the corresponding quadrant of the merged image according to the specified rules.

If necessary, fill the specified color into other areas of the merged image (such as areas not covered by the original image or the reduced image).

Finally,I write the merged image to a file or display it on the screen.