BMP Image Processing

Write a program in C (not C++) that can read a BMP image file, modify the image, and save the modified version as a new BMP file.

Requirements

1. The program must use only standard C libraries such as stdio.h, stdlib.h, and math.h. Third-party image processing libraries like OpenCV are not allowed.

2. BMP (Bitmap) files come in multiple formats, primarily differing in color depth, compression methods, and header versions. The program should focus on handling 24-bit uncompressed BMPs, as they are the most common and easiest to work with.

3. A `struct` should be defined to store the image data in memory after loading it from the BMP file.

4. The program should support basic image processing operations, such as:

   - Adjusting brightness by adding a fixed value to all pixel values (e.g., `$ ./bmpedit -i input.bmp -o output.bmp -op add 50`).
   - Blending two images by averaging their pixel values (e.g., `$ ./bmpedit -i input1.bmp -i input2.bmp -o output.bmp -op average`).
   - Some other image processing functions.

5. Image processing is time consuming. Please optimize the source code to make it running fast.

6. The program should be **robust** (with good memory management, and handling errors gracefully) and user-friendly (with clear command-line usage).

7. You are **encouraged** to use an AI tool, such as DeepSeek, to generate a framework of the source code, and help improve the quality of your source code. Please describe how you improve your source code and your report using the tool. But do not generate some rubbish words and source code.

Rules:

1. The project report and the source code must be submitted before the deadline. Any submission after the deadline (even by 1 second) will result in **a score of 0**. The deadline is 23:59 on April 13.

2. The files should be submitted as report.pdf, *.c (it's better to have one *.c file). Use exact filename capitalization and extensions. The files should **NOT** be compressed into one.

3. The score will depend on the quality of both the source code and the report. The report should be easy to understand and provide a clear description of the project, especially the highlights.

4. Attention should be paid to code style. Adequate time is given for code to be written correctly and with good style. Deductions will be made for poor code style. Code style guides, such as the Google C++ Style Guide (http://google.github.io/styleguide/cppguide.html), can be used as a reference.