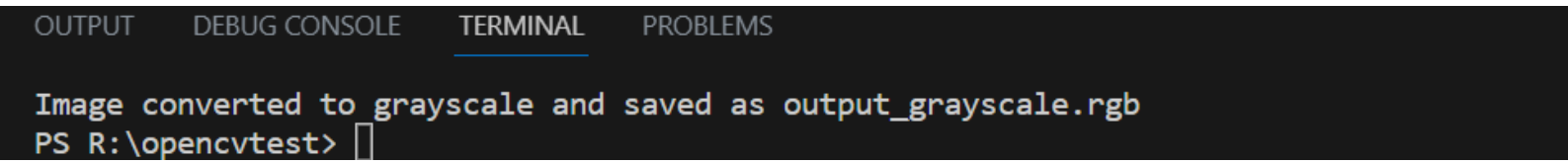
```cpp
#include <iostream>
#include <fstream>
#include <vector>
struct RGBColor {
    unsigned char r, g, b;
};
unsigned char rgbToGrayscale(const RGBColor& color){
    return static_cast<unsigned char>((color.r + color.g + color.b) / 3);
}

int main() {
    std::ifstream inputFile("Photo.jpeg", std::ios::binary);
    if (!inputFile) {
        std::cout << "Could not open or find the image." << std::endl;
        return -1;
    }
    int width = 0, height = 0;
    inputFile.read(reinterpret_cast<char*>(&width), sizeof(int));
    inputFile.read(reinterpret_cast<char*>(&height), sizeof(int));

    std::vector<RGBColor> imagePixels(width * height);
    inputFile.read(reinterpret_cast<char*>(imagePixels.data()),
sizeof(RGBColor) * width * height);
    std::vector<unsigned char> grayscalePixels(width * height);
    for (int i = 0; i < width * height; ++i) {
        grayscalePixels[i] = rgbToGrayscale(imagePixels[i]);
    }
    std::ofstream outputFile("output_grayscale.rgb", std::ios::binary);
    if (!outputFile) {
        std::cout << "Could not create output file." << std::endl;
        return -1;
    }
    outputFile.write(reinterpret_cast<const char*>(&width), sizeof(int));
    outputFile.write(reinterpret_cast<const char*>(&height), sizeof(int));
    outputFile.write(reinterpret_cast<const char*>(grayscalePixels.data()),
sizeof(unsigned char) * width * height);
    std::cout << "Image converted to grayscale and saved as
output_grayscale.rgb" << std::endl;
    return 0;
}
```


OUTPUT
DEBUG CONSOLE
TERMINAL
PROBLEMS
Image converted to grayscale and saved as output_grayscale.rgb
PS R:\opencvtest>