C++ Coding Challenge 2

Implement a light version of our Rendering Engine. Design and implement two classes: `Image` and `RenderingEngine` with just the C++ standard library (no hardware acceleration required). `Image` represents a 32-bit RGBA bitmap in physical memory and will be used in the API of the rendering engine class. The rendering engine has an internal framebuffer which is initially transparent and whose dimensions are set in the constructor. The rendering engine should have the following capabilities (designing its API is part of the task):

- load and save `Image` objects in `.rgba` format: https://bzotto.medium.com/introducing-the-rgba-bitmap-file-format-4a8a94329e2c
- copy the framebuffer to an `Image`
- draw and properly alpha-blend an axis-aligned rectangle with a given position, size, fill color, stroke color, and stroke width
- draw and alpha-blend an image into the framebuffer
    - it should be possible to draw the image into a rectangle with a different size than the original thus resizing the image - use a proper resampling algorithm

Example

Consider the following images:

image-A.rgba

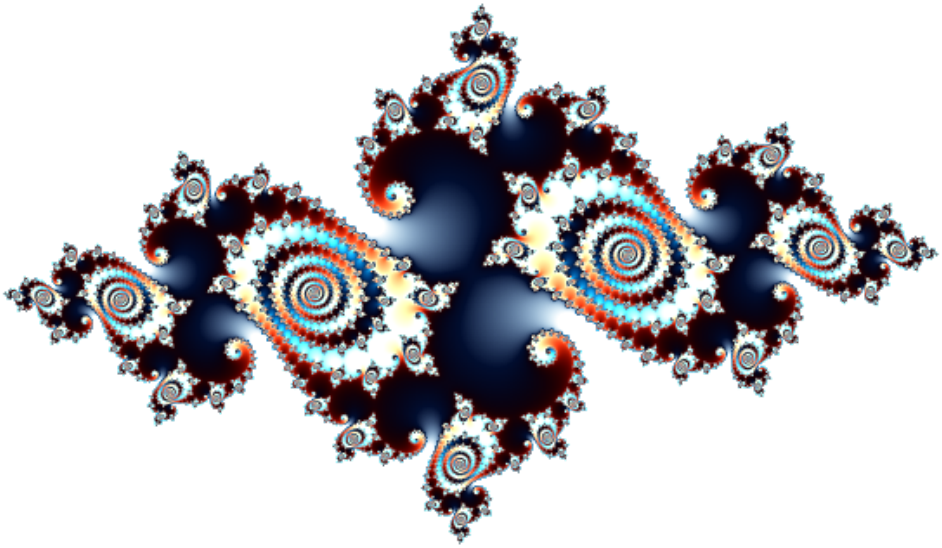
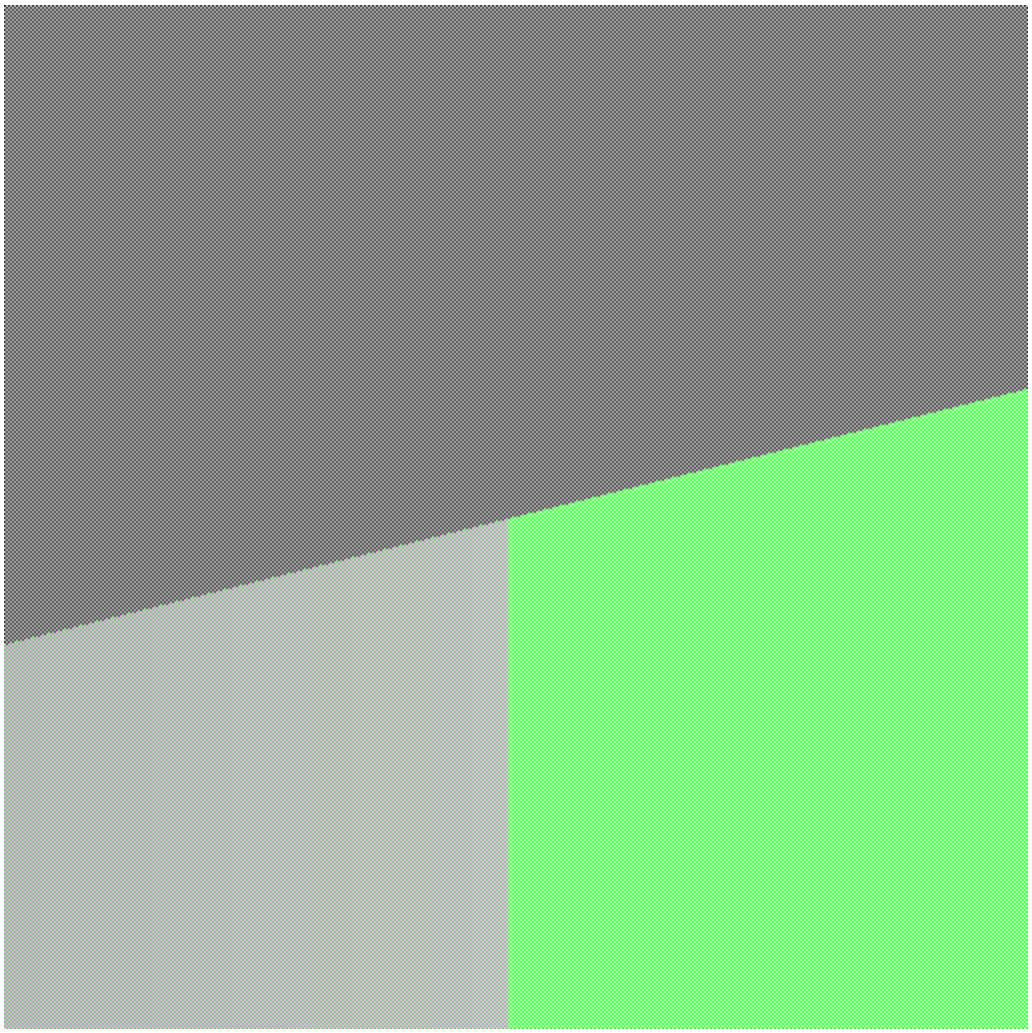


image-B.rgba

image-C.rgba



After sending the following commands to the rendering engine:

- set framebuffer width: 640, height: 480
- draw image-A at left: -40, top: 60, width: 720, height: 360
- draw rectangle at left: 8, top: 8, width: 624, height: 464 with fill color #0000FF, alpha 12.5%, with inner stroke width 24, color #004040, alpha 100%
- draw image-B at left: 0, top: 0, width: 640, height: 480
- draw image-C at left: 0, top: 0, width: 256, height: 256
- copy framebuffer to output-image

You should get output-image.rgba: