1. Introduction

This document describes the testing approach and process used to verify the functionality of the Image Utility application. The purpose of testing is to ensure that each feature of the program works as expected and produces correct outputs for valid inputs, while handling invalid inputs gracefully.

Both **automated tests** (via a dedicated test_all.c program) and **manual interactive testing** were performed. Screenshots of console sessions and sample input/output images are included to illustrate the results.

---

2. Testing Approach

Our testing approach combined the following methods:

1. **Unit Testing**
    - Each function (e.g., load_bmp, save_bmp, fill_bmp, rotate_bmp, etc.) was tested individually.
    - The automated test program uses assert() statements to compare actual outputs with expected results.
2. **Integration Testing**
    - Full application workflows (e.g., load → rotate → save) were executed interactively using the console interface.
    - Commands and outputs were recorded to verify correctness.
3. **Regression Testing**
    - After fixing bugs, tests were rerun to confirm that changes did not break existing features.

---

3. Automated Testing with test.c

A dedicated test file (test.c) was written to exercise all major functions of the application. The test suite includes:

- **Loading and Saving**
    - Verify that BMP files can be loaded into memory.
    - Save the image to disk and confirm that the output file exists.
- **Fill Operation**

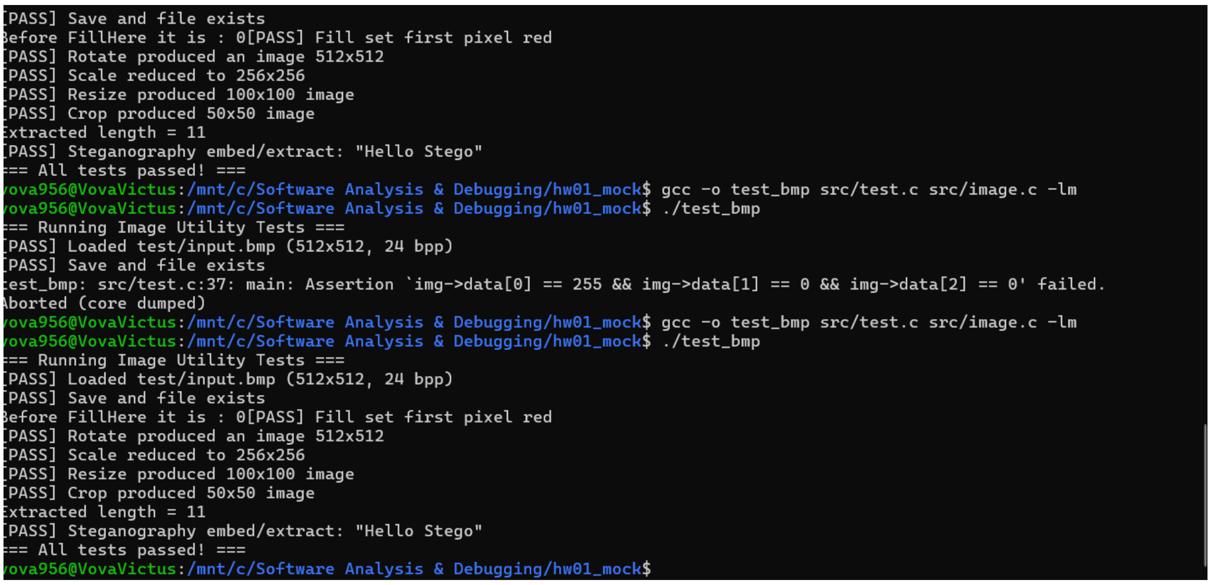
    - Fill the image with a solid color (e.g., red).
    - Check the pixel values to confirm that the image data matches expectations.

- **Rotate, Scale, Resize, Crop**
    - Apply geometric transformations.
    - Verify that image dimensions change correctly (e.g., scaling by 0.5 halves the dimensions, cropping produces the expected width/height).
- **Steganography (Embed & Extract)**
    - Embed a secret message inside the BMP.
    - Extract the message and confirm that it matches the original input.
- **Error Handling**
    - Attempt invalid operations (e.g., scaling by zero, cropping outside image bounds).
    - Confirm that the program fails gracefully with error messages.

---

4. Sample Automated Test Output

```
=== Running Image Utility Tests ===

[PASS] Loaded test/input.bmp (512x512, 24 bpp)[PASS] Save and file exists

Before FillHere it is : 0[PASS] Fill set first pixel red

[PASS] Rotate produced an image 512x512

[PASS] Scale reduced to 256x256

[PASS] Resize produced 100x100 image

[PASS] Crop produced 50x50 image

Extracted length = 11

[PASS] Steganography embed/extract: "Hello Stego"

=== All tests passed! ===
```



```
[PASS] Save and file exists
Before FillHere it is : 0[PASS] Fill set first pixel red
[PASS] Rotate produced an image 512x512
[PASS] Scale reduced to 256x256
[PASS] Resize produced 100x100 image
[PASS] Crop produced 50x50 image
Extracted length = 11
[PASS] Steganography embed/extract: "Hello Stego"
=== All tests passed! ===
vova956@VovaVictus:/mnt/c/Software Analysis & Debugging/hw01_mock$ gcc -o test_bmp src/test.c src/image.c -lm
vova956@VovaVictus:/mnt/c/Software Analysis & Debugging/hw01_mock$ ./test_bmp
=== Running Image Utility Tests ===
[PASS] Loaded test/input.bmp (512x512, 24 bpp)
[PASS] Save and file exists
test_bmp: src/test.c:37: main: Assertion `img->data[0] == 255 && img->data[1] == 0 && img->data[2] == 0' failed.
Aborted (core dumped)
vova956@VovaVictus:/mnt/c/Software Analysis & Debugging/hw01_mock$ gcc -o test_bmp src/test.c src/image.c -lm
vova956@VovaVictus:/mnt/c/Software Analysis & Debugging/hw01_mock$ ./test_bmp
=== Running Image Utility Tests ===
[PASS] Loaded test/input.bmp (512x512, 24 bpp)
[PASS] Save and file exists
Before FillHere it is : 0[PASS] Fill set first pixel red
[PASS] Rotate produced an image 512x512
[PASS] Scale reduced to 256x256
[PASS] Resize produced 100x100 image
[PASS] Crop produced 50x50 image
Extracted length = 11
[PASS] Steganography embed/extract: "Hello Stego"
=== All tests passed! ===
vova956@VovaVictus:/mnt/c/Software Analysis & Debugging/hw01_mock$
```