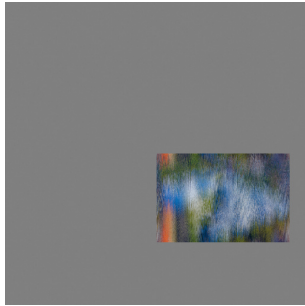


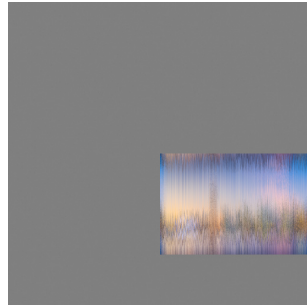
20875 Software Engineering – Assignment 1

Due Monday, October 5th, 2026, 23:59 CEST

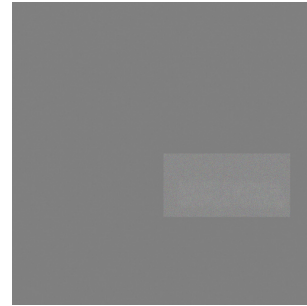
You are given three images:



s_building.png



s_city.png



s_noisy_text.png

Scaled down versions of the three images; Download the full images at
<https://www.poirrier.ca/permanent/images.zip>

All three images have been scrambled with the same method. The assignment consists in implementing a program that, given any image scrambled with this method, will recover the original image exactly. Note that you are not given the original images for the three examples. You are expected to infer the scrambling method from the scrambled images only. The method may involve pseudorandom numbers, and only supports original images of certain sizes. It always produces scrambled images of the same fixed size, in the PNG format.

Grading will be partially automated, and you are asked to follow precisely the instructions given in this document. In particular, your program will be invoked in a specific way detailed below. The assignment is to be uploaded as a single archive file (either `.zip` or `.tgz`) on BlackBoard by the end of November 5th. The archive contains your source code. Do not include images.

Rules. This is an individual assignment. You are allowed to talk about the assignment with your classmates. However, you must write all your code on your own. As a consequence, upon request, you will be able to explain and modify any part of your code. The use of LLMs is explicitly allowed for this assignment, but keep in mind that they may or may not be useful.

Invocation. This assignment can be completed in Python or in C. For any other language, you must get approval first. The program will be invoked with the following commands:

Python: `python3 unscramble.py {input.png} {output.png}`

C: `./unscramble {input.png} {output.png}`

where `{input.png}` is the path to an existing scrambled PNG file and `{output.png}` is the name of the output PNG file to be produced.

External libraries/modules. You are allowed to use anything included in the standard library of the programming language you use. In addition you are allowed to use Python's PIL module (Pillow), and the C libraries `libpng`, `libspng`, and `stb_image`. Other libraries are allowed only if you get approval first.

Compilation. [Only for assignments completed in the C language.] If a file called “Makefile” is present in the submitted archive, the code will be compiled with the command: `make`
Otherwise, it will be compiled with the command `clang -Wall -O3 -o unscramble *.c -lpng -lspng`

Grading.

- [4 marks] When given any image scrambled with the same method that produced `s_building.png`, `s_city.png` and `s_noisy_text.png`, the program successfully recovers the original image.
- [1 mark] When passed an invalid input image (e.g. an image that was not scrambled), the program is allowed to produce an invalid output or no output at all. However, even in this case, the program must not crash and must not report any uncaught exception.

Note: This homework assignment serves as a preparation for the first in-class assignment, which will cover a related topic.

Checklist

A few quick checks to prevent common and easily avoidable mistakes:

- ☐ Assignment uploaded as a single `.zip/.tgz`
- ☐ [Python] Main script is called `unscramble.py`
[C, if a Makefile is provided] Executable built by `make` is called `unscramble`
- ☐ Program takes the correct two arguments as specified (input file and output file)
- ☐ [Python] Script does not abort with uncaught exceptions, even when the input is invalid
[C] Executable does not crash, even when the input is invalid