

COM2113 Fall 2022, Assignment 3

Mandelbrot Renderer, Part 2

In this assignment, you will be writing in Assembly a function to calculate the escape time of a particular location in the fractal plane.

[Optional background reading on the Mandelbrot Set](#)

You will be implementing this “[Unoptimized naïve escape time algorithm](#)” in assembly. I have provided two files:

- MBPixelCalc.s – A sample assembly file with the function MBPixelCalc. This sample returns the value 24.
- tester.c – A sample C file that calls the function MBPixelCalc and uses printf() to print the value returned by MBPixelCalc.

To compile, use **gcc tester.c MBPixelCalc.s -o tester** to compile the executable “tester”. You then type **./tester** to execute it.

As in the last assignment, we will be using a fixed-point datatype for all calculations. The datatype is a 64-bit signed integer, where we consider the 6 most significant bits to be to the left of the binary point. I refer to this type below as “6.58”, meaning 6 bits are to the left of the binary point, and 58 bits are to the right of the binary point. This provides a range from -16 to 16-ε.

Your tester.c, when executed, takes from the command line the x and y value to be tested. Tester.c will read these two values (for example, 1.652, -0.0427) into the type **double**, then convert these two values into our custom “6.58” type and insert the appropriate values into a pair of **longs** that are passed to MBPixelCalc.

MBPixelCalc will then run the algorithm linked above using the 6.58 datatype. It will return value of **iteration** from the algorithm above.

You may use as resources the textbook and slides and lectures, as well as links provided in the slides. Anything else is probably cheating. Please e-mail me if you have questions about this.

Hand-in will be a zip file via Canvas, containing our tester.c and MBPixelCalc.s.