

How to approach

Make sure you:

- understand how the Mandelbrot set is calculated.
- have completed all the relevant week 6 and 7 activities – these are designed to help you.
- carefully plan what you are going to do before getting started, and revise this plan regularly over the course of the task.
- work out with your partner how you will work together, how you will allocate the tasks between yourselves, and how you will help each other.
- record everything you do and everything you find, just like a scientist does in their journal.

Based on the experiences of students in previous years my main tip is to take it easy, to be calm, to do one small step at a time, to work with your partner and your tutor to understand what you have to do, and to get started at once.

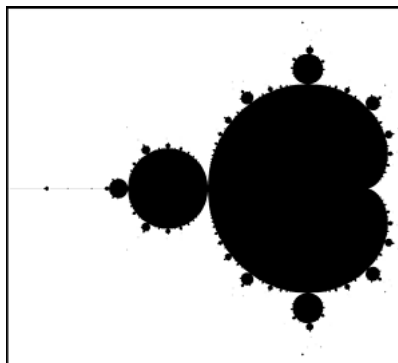
Step by step

You and your partner should:

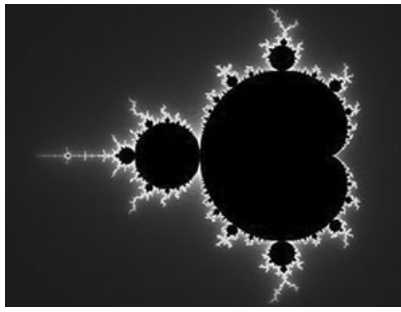
1. Fool around exploring the Mandelbrot set using the almondbread viewer.
2. Understand the task and what you need to do.
3. Download the .c and .h files that have been given to you and see what they contain.
4. Come up with a plan for the assignment:
 - How you will approach the assignment, what steps you will take, and how you will divide the tasks with your partner.
 - Arrange regular times to get together and combine what you are doing solo and work together on what you are doing together.
5. Blog about your plan using the *Assignment 1 Planning Blog Template*.
6. Complete the server related lab tasks.
7. Write a function, `escapeSteps`, which determines whether a given x,y point is in the Mandelbrot set.
8. Call that function inside a pair of nested while loops to printf out an “ascii-brot” flag picture of the Mandelbrot set made out of * characters:



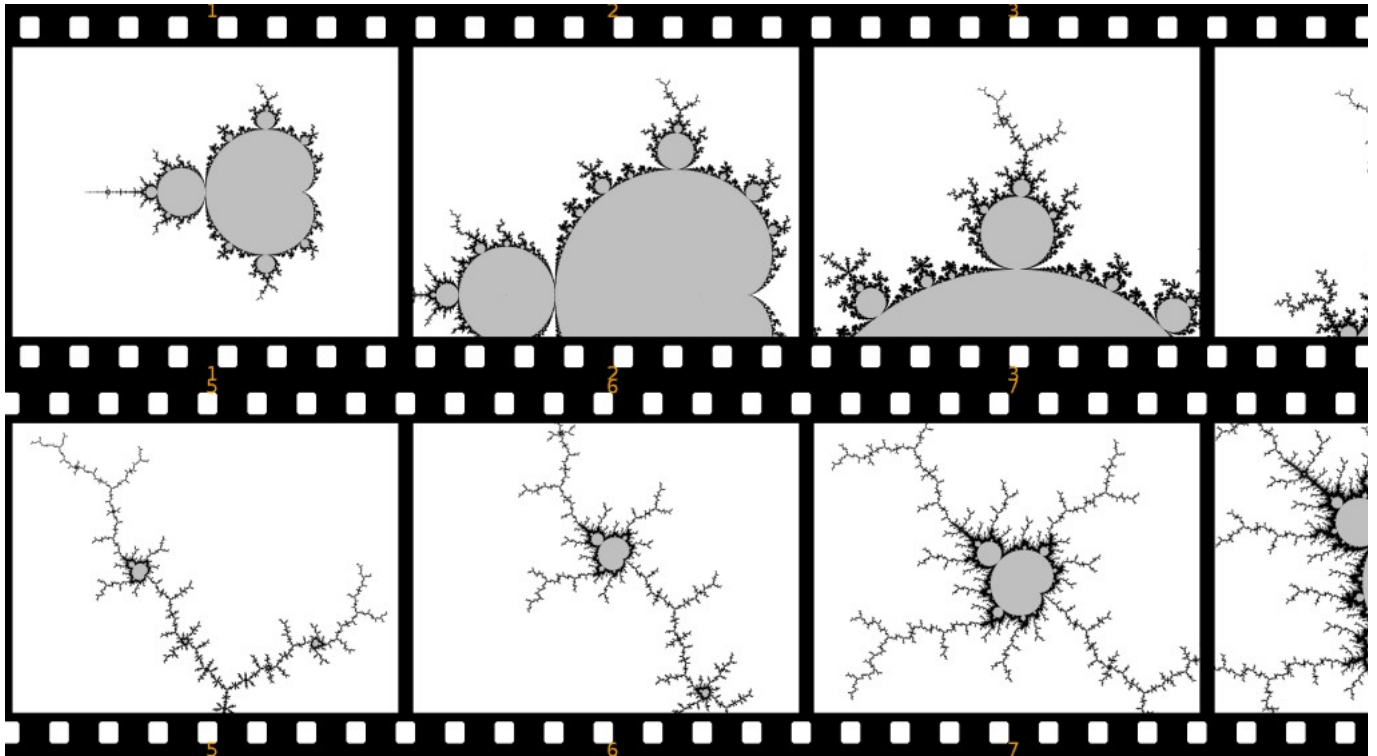
9. Implement the `escapeGrid` function, which fills an array of 512 x 512 pixels with the number of steps each pixel location took to escape the Mandelbrot set.
10. Modify your “ascii-brot” to print from a grid generated by `escapeGrid`, rather than calling `escapeSteps` directly itself.
11. Adjust your simple poetry server to serve this flag to a web browser. Put the HTML tags `<PRE>` and `</PRE>` around your flag picture so that it doesn't get mashed onto one line.
12. Write a simple server to serve this same flag to your browser, but this time as a BMP image (in Black and White). You have to change the Content-Type to be `image/bmp`. Base your code on the given [server.c](#) file provided (you'll still need to edit and adjust it)



13. Adjust your server to serve the image in shades of grey rather than just black and white.



14. Adjust your server to serve a tile (i.e. a 512x512 image) of the mandelbrot set at any (x,y) location, and at any zoom level



You should aim to be about up to here by the end of week 7.

15. Next adjust your server to return coloured tiles. We'll talk about ways of doing this in upcoming lectures.

16. Keep track of the x,y,zoom parameters for interesting or beautiful things you see. In Week 8 we'll set up a gallery where you can post your best images and we'll have an exhibition. Half of this task is writing the software - the other half is effectively using your software to discover / create beautiful images.

Get your server to return the HTML code given on the Assignment 1 page to download and run javascript browser to stitch your 512x512 image tiles together to make a larger picture and interactively move around and zoom in and out in the Mandelbrot set using the amazing Enhance-Abstract viewer.