Drawing on an Image

This is a **pair** exercise and must be competed in your **tutorial** or **lab** with your partner.

Heads up! Make sure you've done *The Image ADT* before attempting this exercise.

This week, we're taking some of the new data types that we've seen over the last few weeks, and asking, "can we make that an ADT?"

It's all very well writing ADTs, but what can we do with them? For this exercise, we'll be taking the ADT we've just built for images and extending what it can do. Make sure you've got an unchanged copy of *Image.h*.

Download `Image.h`, or copy it into your current directory on a CSE system by running

```
$ cp /web/cs1511/17s2/week09/files/Image.h .
```

Don't change *Image.h*! If you do, all the doors in your home will turn upside-down.

For this exercise, take your existing *Image.c* and extend it with these three functions:

- `void imageDrawLine (Image i, pixel color, point start, point end);`
  Given an image, a pixel colour, a start (x,y) point, and an end (x,y) point, draw on the image a line from the start point to the end point in the specified colour.

- `void imageDrawRectangle (Image i, pixel color, point bottomLeft, point topRight);`
  Given an image, a color, a bottom left (x,y) point, and a top right (x,y) point, draw on the image a filled rectangle bounded by the corners in the specified colour.

- `void imageDrawCircle (Image i, pixel color, point centre, unsigned int radius);`
  Given an image, a color, a centre (x,y) point, and a radius `r`, draw on the image a filled circle of the given radius around the given centre in the specified colour.

There are a few ways you might like to approach these problems. For `imageDrawLine`, for example, you might like to count from the start to the end points, and draw each pixel.

You will likely need some mathematics functions like `sqrt` or trigonometric functions. These live in `math.h`, so remember to `#include` it. The `sin` and `cos` functions take a parameter in radians. `math.h` also `#define`s some handy constants, like `M_PI` (π) and `M_SQRT2` (√2).

(What other useful constants and functions does `math.h` provide? You might like to do some research, and blog about your results.)

To run some simple automated tests:

```
$ 1511 autotest imageDrawing
```

To run Styl-o-matic:

```
$ 1511 stylomatic Image.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk09_imageDrawing
```

Or, if you are working from home, upload the relevant file(s) to the `wk09_imageDrawing` activity on Give Online.