Reticulating Splines

This is a **challenge** exercise. It is **not compulsory**, and may be completed **individually or with your lab partner**.

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

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

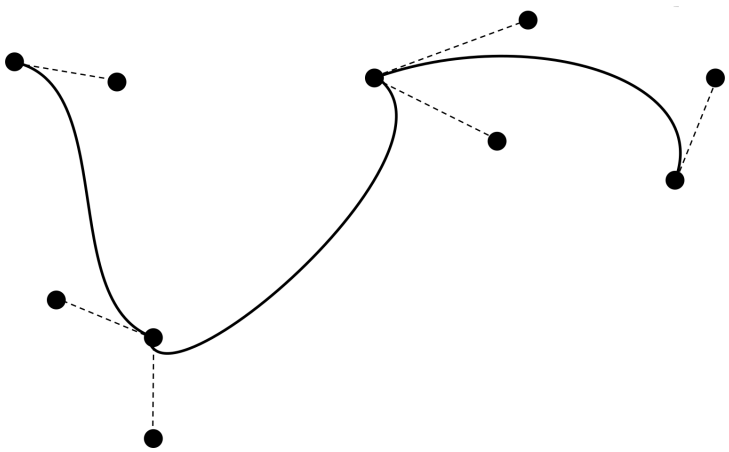
```
$ cp /web/cs1511/17s2/week09/files/Image.h .
```

Don't change *Image.h*! If you do, weasels will eat your phone system.

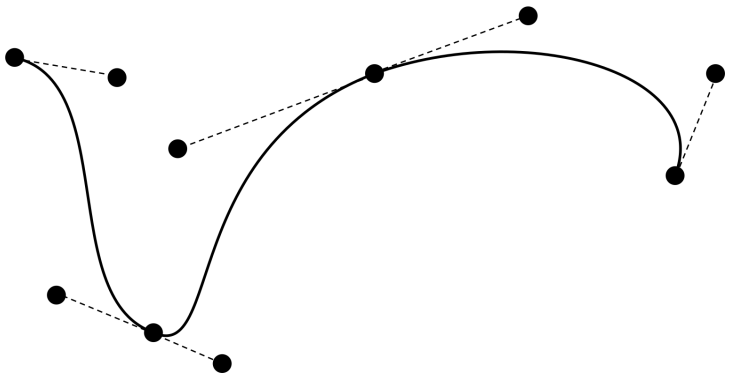
A **Bézier curve** is a parametric curve bounded and shaped by a set of *control points*. A Bézier curve of order 2 has three control points, and is a parametric parabola; a Bézier curve of order 3 has four control points, and is a parametric cubic.

A curve won't escape from the *bounding box*, the closed shape of the control points. For an order-3 Bézier curve, that's usually a quadrilateral.

A sequence of Bézier curves where the end-point of one is the start-point of another forms a *spline*, and the points where they join are called *knots*.



If the series of control points are collinear – that is, a line can be drawn through them and through the knot – the knots have continuity.



Bézier curves and splines are drawing primitives, and are especially important in vector graphics, like the very characters you're reading now.

A Bézier curve is parametric, and the function for an order-3 Bézier curve is, for control points P_0, P_1, P_2, P_3, and for all real values t from 0 to 1,

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3$$

For each t, the actual coördinate is the combination of x values and of y values.

For this exercise, take your existing *Image.c* and extend it with the following function:

- `void imageDrawBezier (Image i, pixel color, double precision, point p0, point p1, point p2, point p3);`
  Given an image, a colour, a "precision", a start (x_0, y_0) point, two control points (x_1, y_1) and (x_2, y_2), and an end point (x_3, y_3), draw on the image a order-3 Bézier curve bounded by those points, in that colour, whose t values increase by the specified precision at each iteration.

To run some simple automated tests:

```
$ 1511 autotest imageDrawing3
```

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_imageDrawing3
```

Or, if you are working from home, upload the relevant file(s) to the `wk09_imageDrawing3` activity on Give Online.