Extra ideas for Assignment 3

Recursion is hard. If you find it difficult to work out what you need to do, that is perfectly normal. Even the best programmers do not usually invent recursive functions from scratch. Here are some pointers on how to approach Assignment 3:

1. The recursive algorithm (and function):

The recursive algorithm for checking the pixels looks like this:

Find pixel A at location (x, y) and look for all connected pixels by:
- Go to pixel B at location (x – 1, y) and look for all connected pixels;
- Go to pixel C at location (x + 1, y) and look for all connected pixels;
- Go to pixel D at location (x, y – 1) and look for all connected pixels;
- Go to pixel E at location (x, y + 1) and look for all connected pixels;

And there are the two base cases.

First thing to note - **this is a void function**. It does things but it does not return any values. The parameters should be x and y (the location of the pixel).

The basic approach to your code for this algorithm is:

```
void findConnectedPixels(int x, int y) {
  if (picture[x][y].getexclude() == true) { return; }  // base case one
  if (picture[x][y].getR() <= 128) { return; }  // base case two

  // use x and y to assist in calculating xmax, xmin, ymax, ymin

  picture[x][y].setexclude(true);   // do not look at this pixel again

  findConnectedPixels (x - 1, y);
  findConnectedPixels (x, y - 1);
  etc...
}
```

2. Drawing a box around a button

To draw a box, you need the values xmin, xmax, ymin, ymax. This is the main work of the recursive function. It keeps check xmin (etc) against x (or y) and adjusting if needed.

Initialise xmin (etc) in main – do not initialise them inside the function.

Draw the box in main – do not draw the box inside the function.