159.233 Assignment 3 - ARM line drawing

Due 3rd June 2015

This assignment is about writing a line drawing subroutine in ARM assembly language.
The following is a C implementation of Bresenham's algorithm for drawing a line between two points.
The line starts at (x0,y0) and ends at (x1,y1), 'fb' is a pointer to the frame buffer and 'colour' is the colour of the line. (0,0) is top left on the screen.

```
void line(char *fb, int x0, int y0, int x1, int y1, int colour) {
    int steep, deltax, deltay, error, ystep, y, x;
    steep = abs(y1-y0) > abs(x1-x0);
    if (steep) {
        swap(x0, y0);
        swap(x1, y1);
    }
    if (x0>x1) {
        swap(x0, x1);
        swap(y0, y1);
    }
    deltax = x1-x0;
    deltay = abs(y1-y0);
    error = deltax/2;
    y = y0;
    if (y0<y1) ystep = 1; else ystep = -1;
    for (x=x0; x<=x1; x++) {
        if (steep) plot(fb,y,x,colour); else plot(fb,x,y,colour);
        error = error-deltay;
        if (error<0) {
            y = y + ystep;
            error = error + deltax;
        }
    }
}
```

On the paper Stream site is a test program (a3-2015.asm) that draws a rotating cube, it needs to call a subroutine named "line". The line subroutine takes 6 parameters in registers R0..R5. R0 is a pointer to a frame buffer of 640x480 pixels, each pixel is a 16 bit value and pixels are stored left to right and top to bottom. R1 and R2 are the x and y coordinates of the start of the line. R3 and R4 are the x and y coordinates of the end of the line and R5 is the 16 bit value to be used for the colour of each pixel on the line.
Your task is to add the line subroutine to the test program so that it works.
To test your program, use the updated gcc distribution available from the web site which includes an arm assembler and emulator.

Hints:
The address of the pixel at (x,y) will be fb+y*1280+x*2 (it's better to use shifts rather than the MUL instruction for this)
Use the STRH instruction to store the colour value for a pixel
Make sure you save any registers you use in your subroutine on the stack.
Start by writing a plot subroutine to draw a single pixel and test it by drawing a pixel at the start of the line.

Marks will be awarded for well written, optimised code that takes advantage of the features of the ARM CPU.
Submit your .asm file on Stream.
Make sure your id is on a comment line at the beginning of your code and use comments for documentation. Marks will be subtracted for plagiarism, late submission and bad documentation.

M Johnson 2015