```
/*  CS 365, 30 Jan 2004 */

/* Draw a line between two points using Bresenham's algorithm */

void draw_line(point p0, point p1, rgb_pixel c) {
   int delta_x = p1.x - p0.x;
   int delta_y = p1.y - p0.y;

   /* ensure delta_x is positive */
   if (delta_x < 0) {
      draw_line(p1, p0, c);
      return;
   }

   /* 0 <= Slope <= 1 */
   if ((0 <= delta_y) && (delta_y <= delta_x)) {
      draw_line_shallow(p0, p1, delta_x, delta_y, 1, c);
      return;
   }

   /* -1 <= Slope < 0 */
   if ((-delta_y <= delta_x) && (delta_y < 0)) {
      draw_line_shallow(p0, p1, delta_x, -delta_y, -1, c);
      return;
   }

   /* 1 < Slope (or vertical) */
   if (delta_y > delta_x) {
      draw_line_steep(p0, p1, delta_x, delta_y, 1, c);
      return;
   }

   /* Slope < -1 (or vertical) */
   if (-delta_y > delta_x) {  /* Note p0.y > p1.y, so draw in reverse order */
      draw_line_steep(p1, p0, delta_x, -delta_y, -1, c);
      return;
   }

   /* If we recognized all the cases this will not happen! */
   printf("Slope not recognized: delta_x=%d delta_y=%d\n", delta_x, delta_y);
   return;
}



/* Draws a line where |slope|<=1 using Bresenham's algorithm by
 * incrementing along the X coordinate
 */
draw_line_shallow(point p0, point p1,
                  int delta_x, int delta_y, int dir, rgb_pixel c) {
   int x, y;
   int eps = 0;
   y = p0.y;
   for (x = p0.x; x <= p1.x; x++) {
      colorit(x, y, c);
      eps +=  delta_y;
      if ((eps+eps) >= delta_x) {
         y += dir;
         eps -= delta_x;
      }
   }
   colorit(p1.x, p1.y, c);
   return;
}

/* Draws a line where |slope|>1 (or vertical) using Bresenham's
 * algorithm by incrementing along the Y coordinate.
 */
draw_line_steep(point p0, point p1,
                  int delta_x, int delta_y, int dir, rgb_pixel c) {
   int x, y;
   int eps = 0;
   x = p0.x;
   for (y = p0.y; y <= p1.y; y++) {
      colorit(x, y, c);
      eps +=  delta_x;
      if ((eps+eps) >= delta_y) {
         x += dir;
         eps -= delta_y;
      }
   }
   colorit(p1.x, p1.y, c);
   return;
}
```