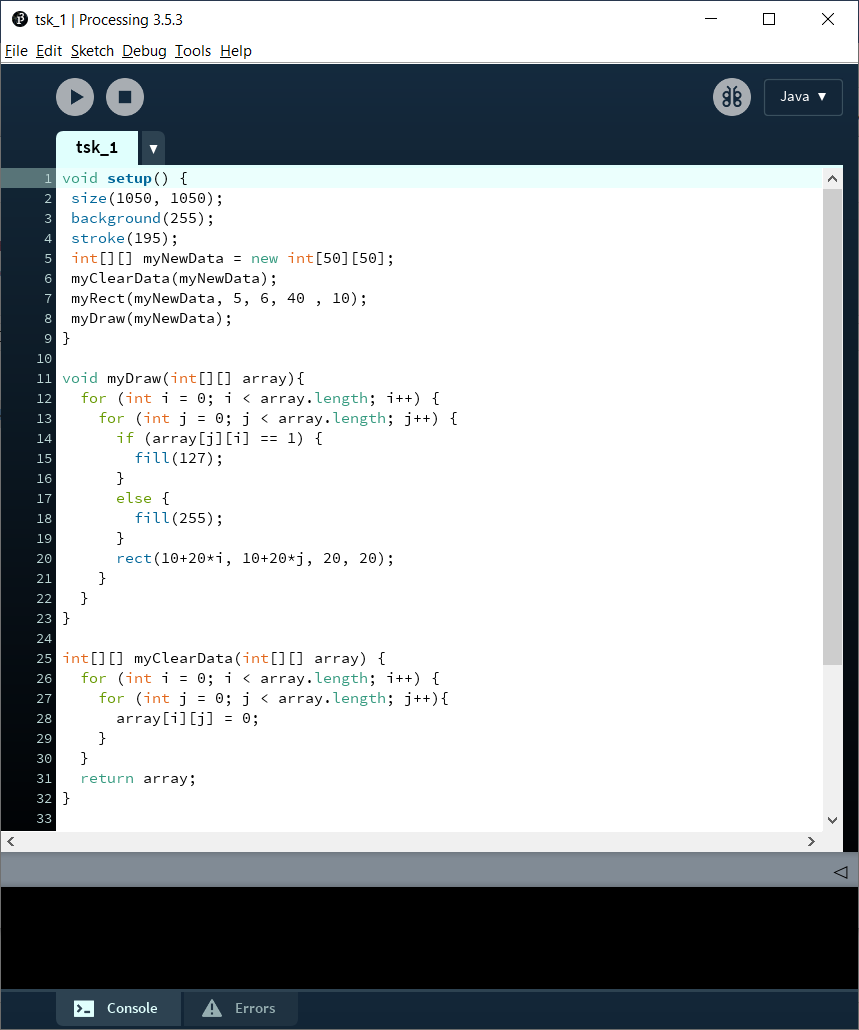
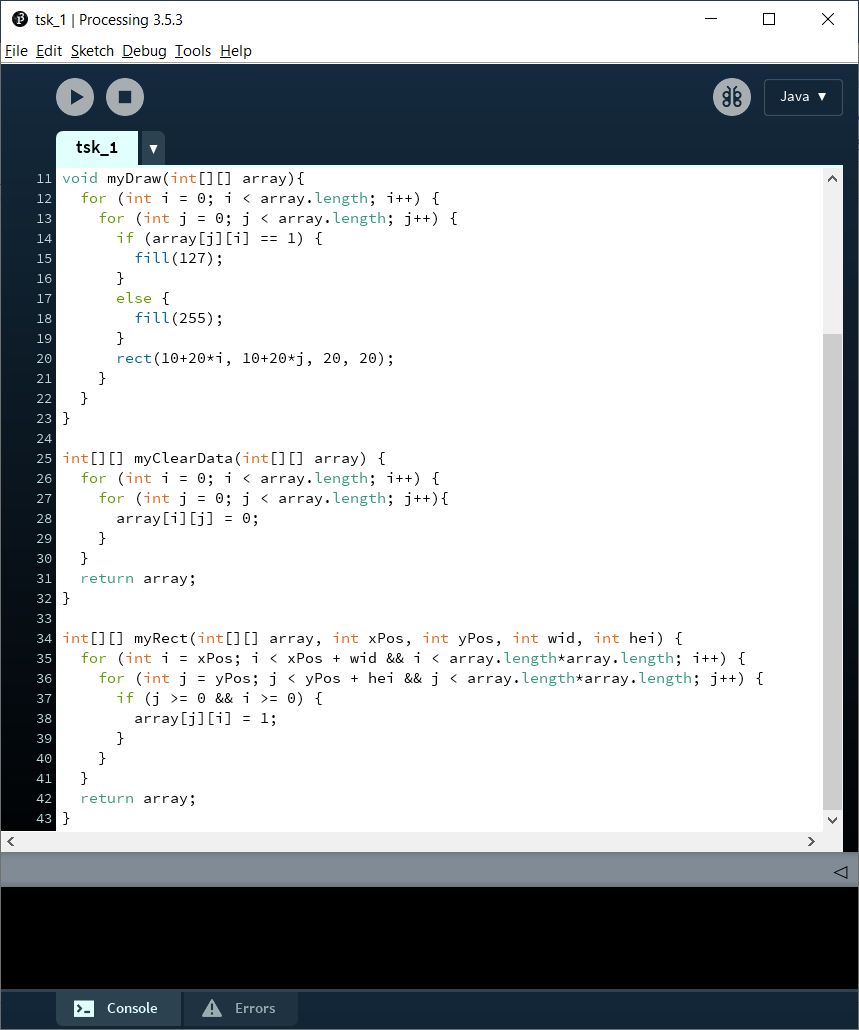
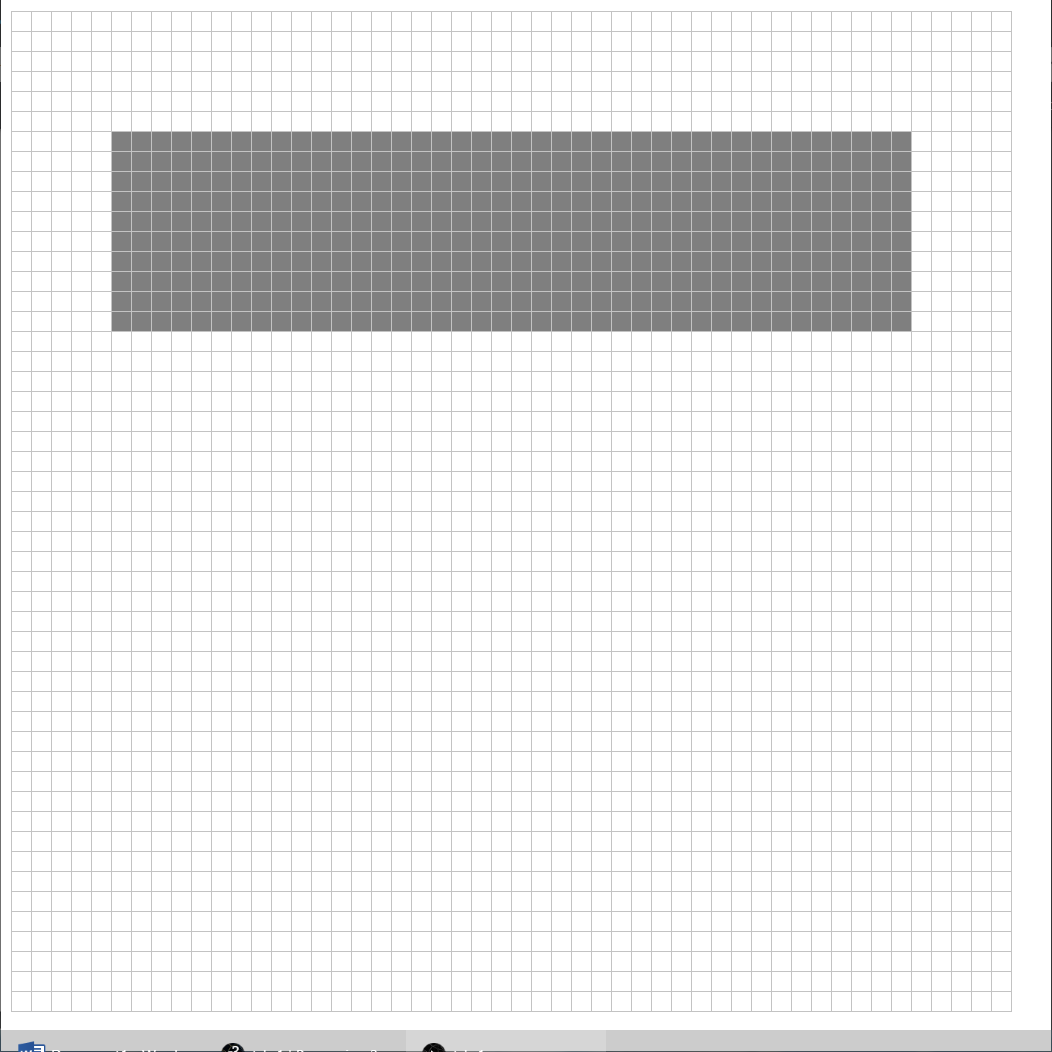
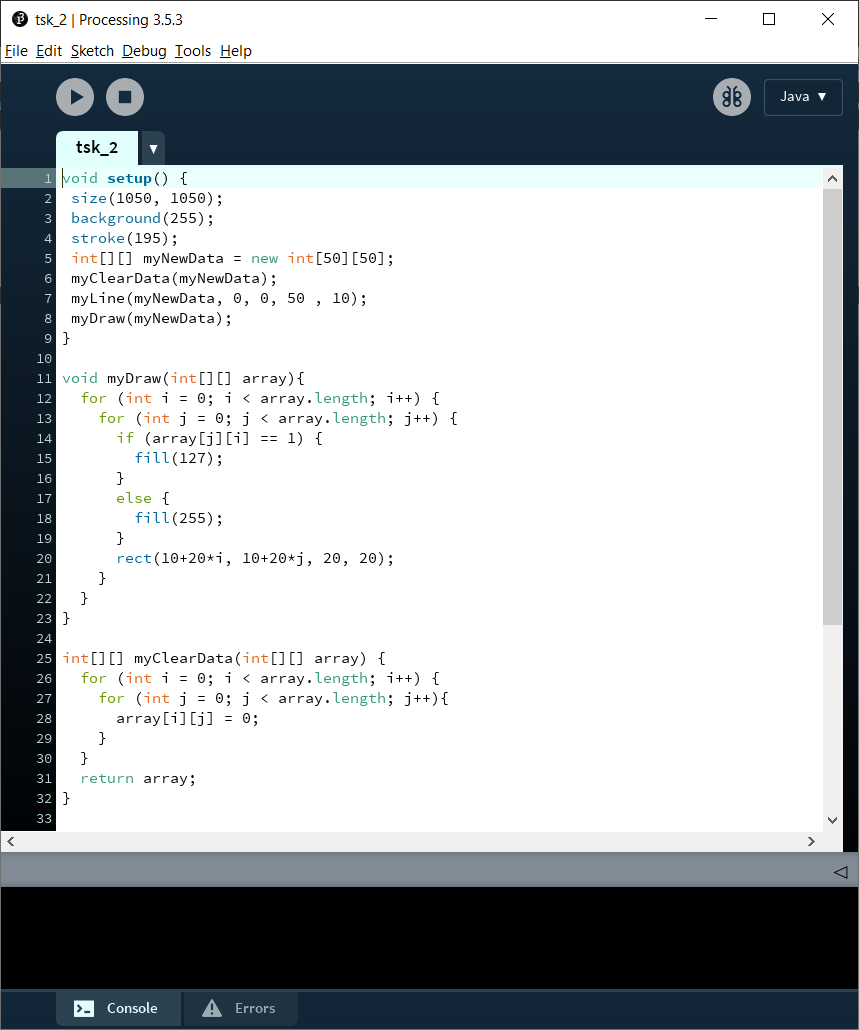
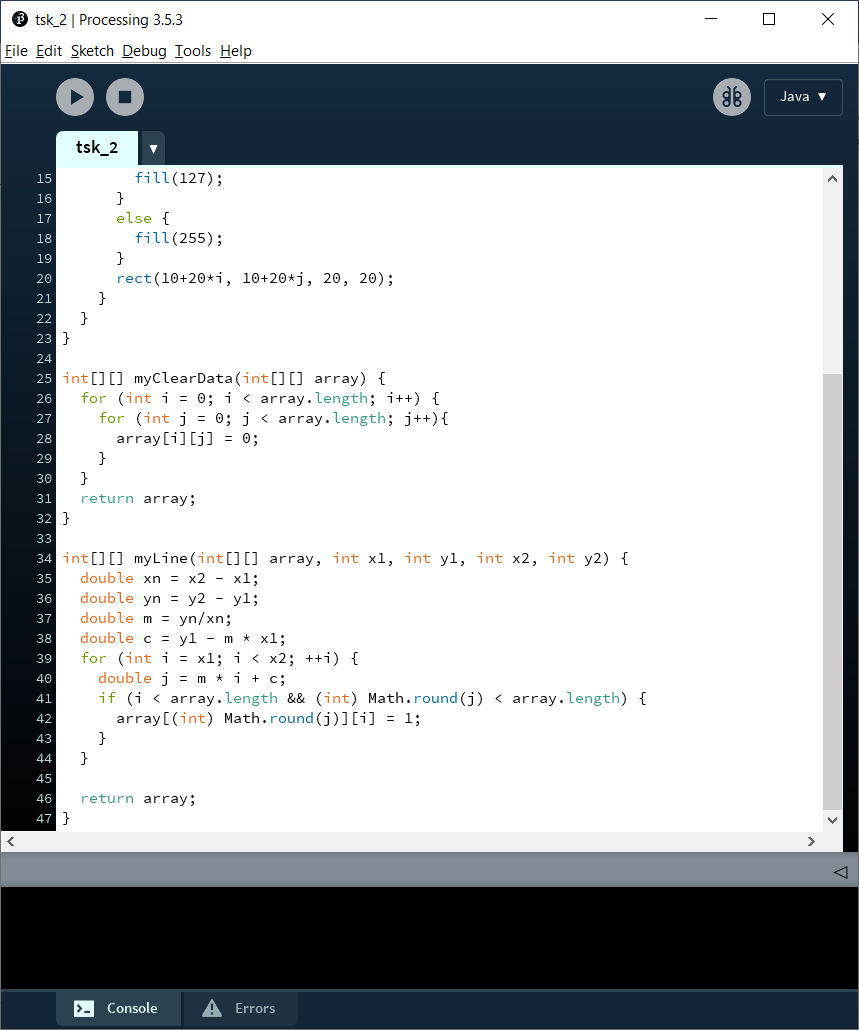
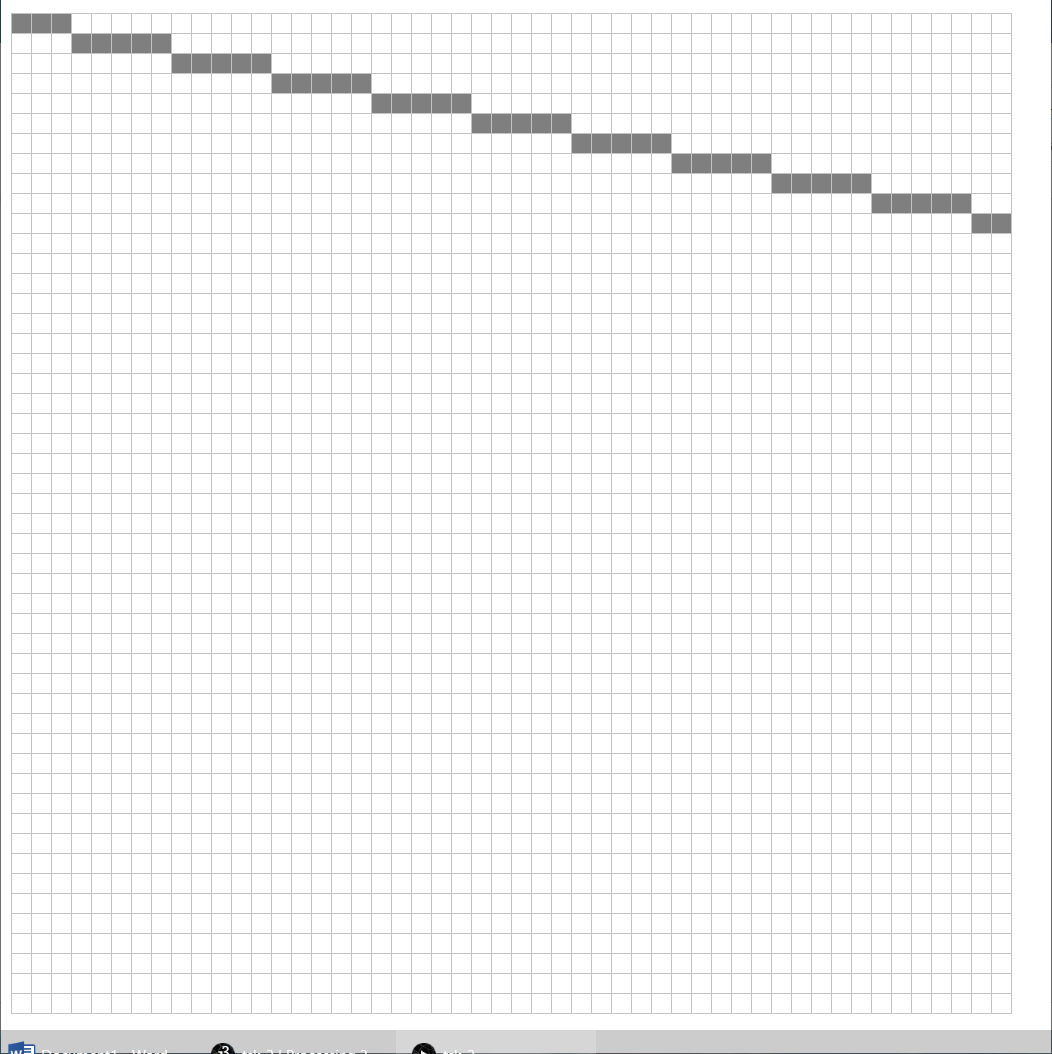
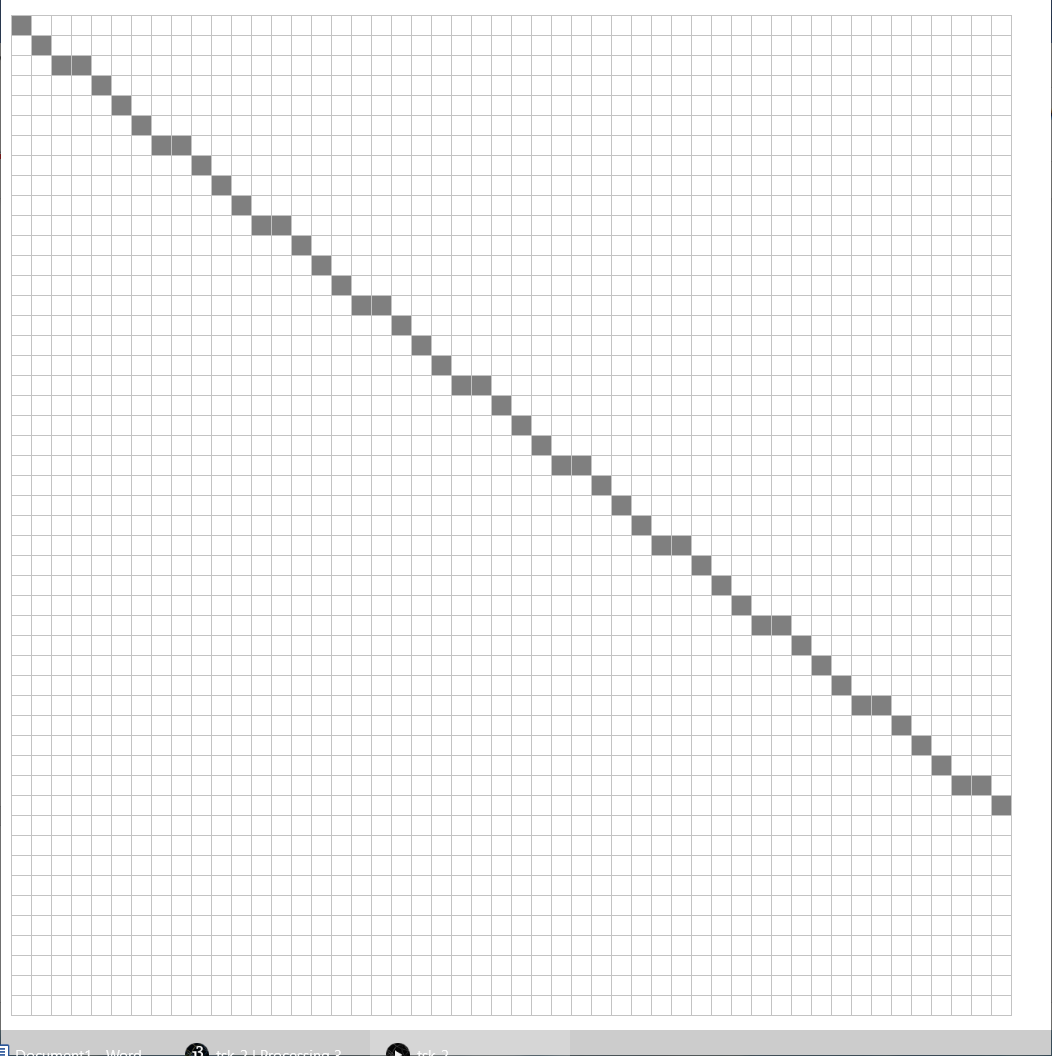
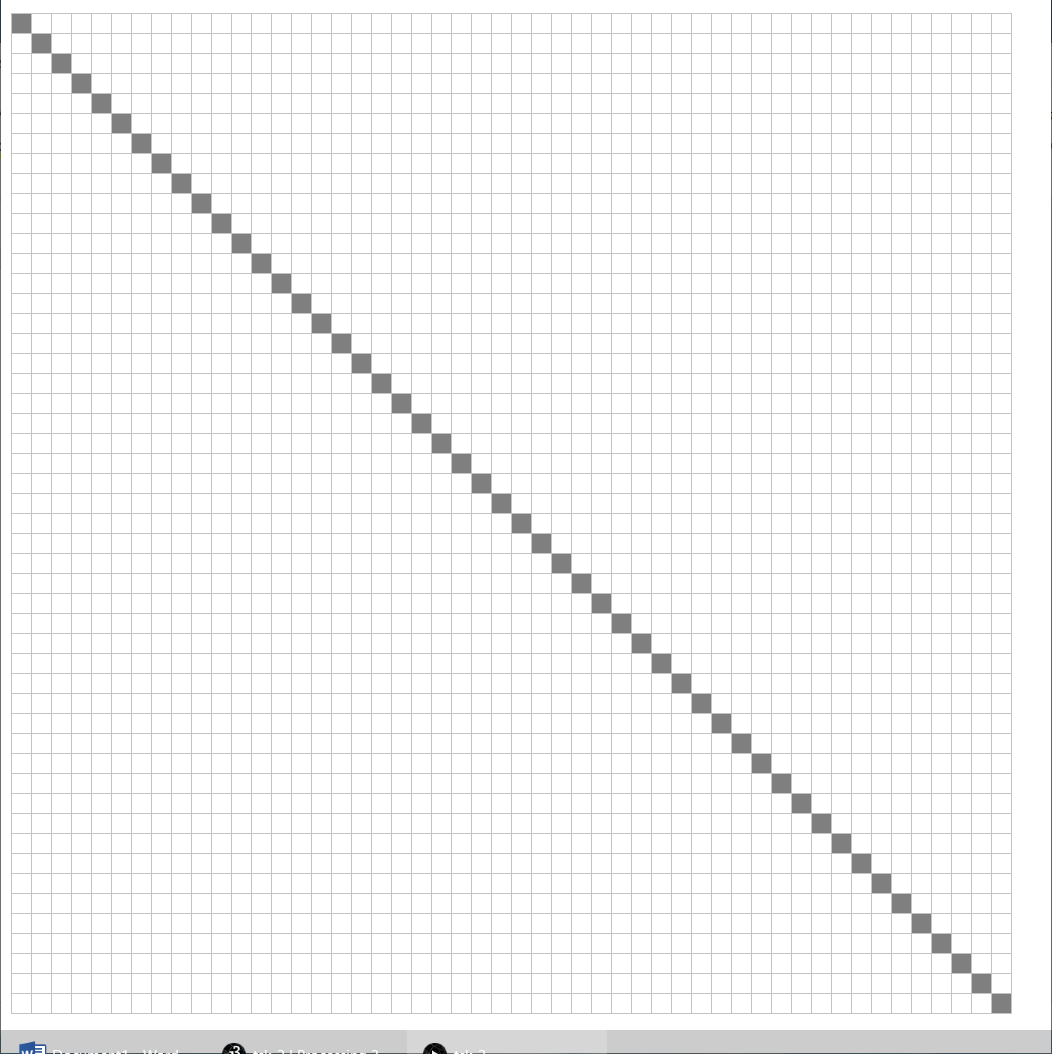
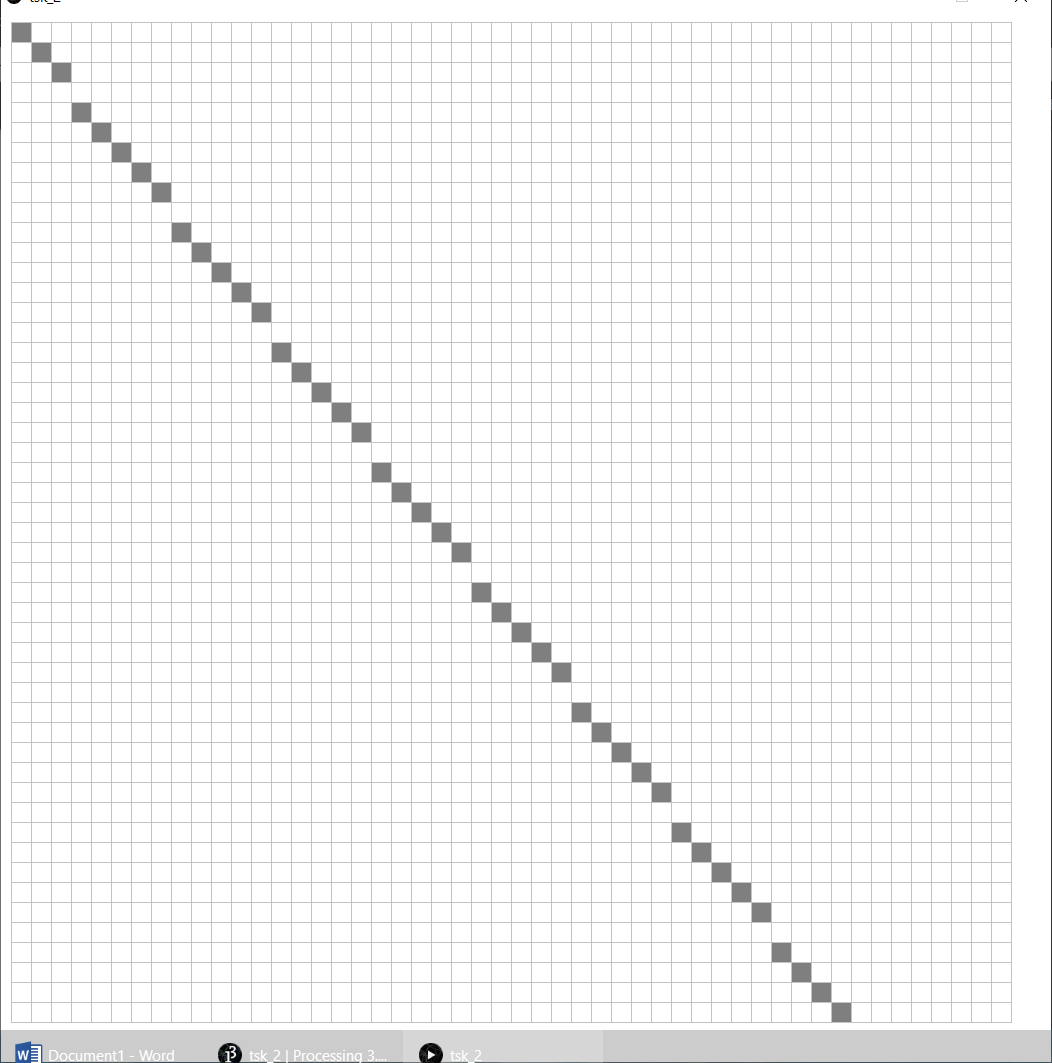
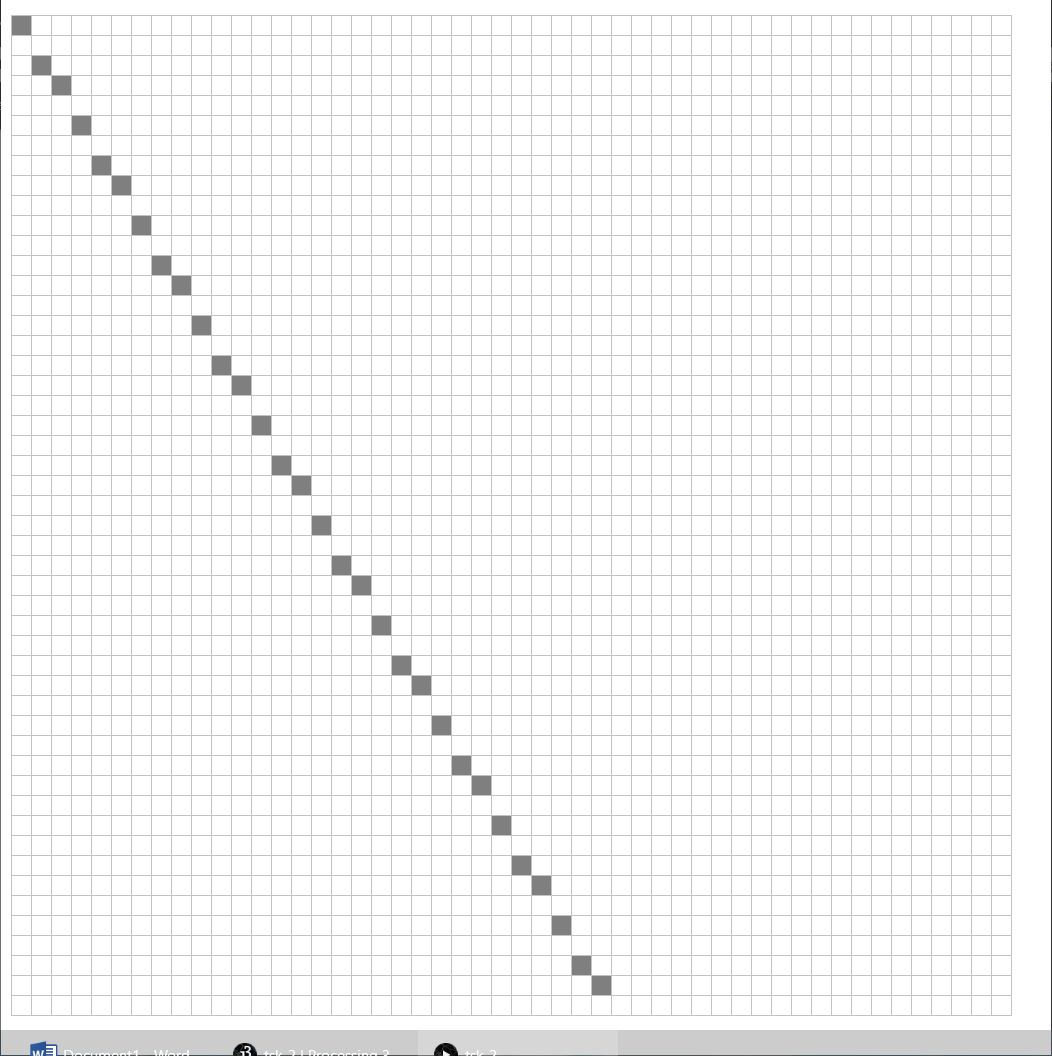
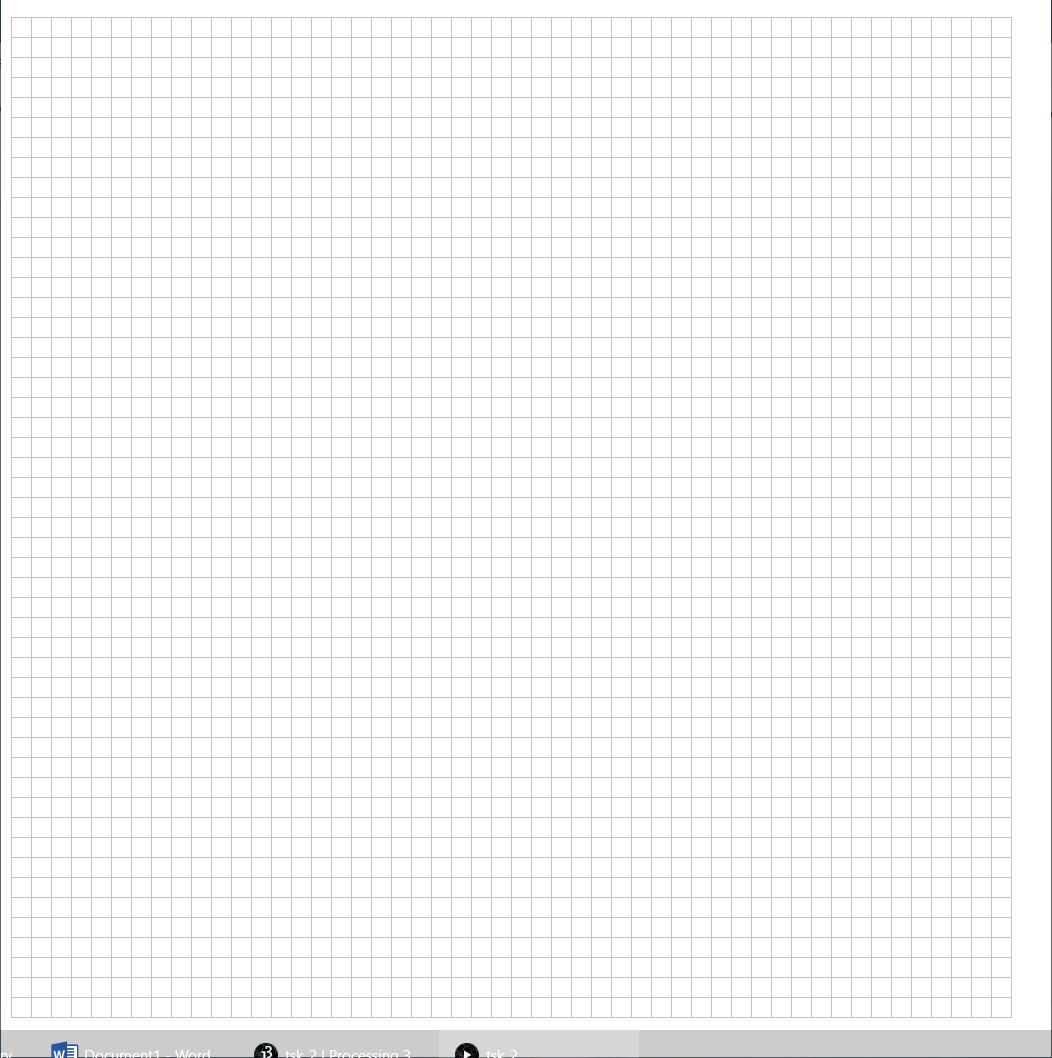
Lab 4

* 1. i) The grid is filled; ii) A square in the bottom right corner of the last 20 squares is filled in; iii) All but the rightmost column is filled in, as it starts at -1 – outside the grid  
       
       
     

1.   
   
   1. A diagonal line is drawn from (0, 0) to (50, 10)  
      
   2. A diagonal line is drawn from (0, 0) to (50, 40)  
      
   3. A diagonal line is drawn from (0, 0) to (50, 50) this is straighter than the others as it has a direct diagonal path between the two coordinates  
      
   4. A diagonal line is drawn from (0, 0) to a point outside the grid. This causes the line to be stretched and so some of the points are missing. This could be fixed by giving the Y-axis calculations a more accurate rounding  
      
   5. A diagonal line is drawn from (0, 0) to (30, 50). Because the Y-axis calculations have a decimal value rounded to the nearest whole number, some of the points are missing from the line. This could also be fixed by giving the Y-axis calculations a more accurate rounding  
      
   6. The line starts drawing at the edge of the grid, but the algorithm used does not account for a line going in a negative direction, so no line is drawn. This could be fixed by determining which point is the smallest, and then drawing from that point  
      
   7. It doesn’t look like a good circle as there are many missing pixels and some erroneous ones. I could improve the circle by giving a more accurate rounding or using a finer/larger grid
   8. The problem with the naïve circle algorithm is that circles are not very well represented with such large pixel sizes – they require finer detail for their curved sides  
      