1. Let \( f \) be given on the square with points \((x, y), (x + h_x, y), (x, y + h_y), (x + h_x, y + h_y)\) and suppose we want to approximate a value at \((x + k_x, y + k_y)\), for \(0 \leq k_x \leq h_x\) and \(0 \leq k_y \leq h_y\). Use 1D linear interpolation to approximate values at \((x + k_x, y)\) and at \((x + k_x, y + h_y)\), then use 1D linear interpolation to approximate the value at \((x + k_x, y + k_y)\). Simplify your results and show it is the same as the usual formula for bilinear interpolation.

2. Resize, using bilinear interpolation, the image of “bird.bmp” or “seal.bmp” or “lizard.bmp” into an image with width 500 and aspect ratio:
   (a) \(4 : 3\)
   (b) \(16 : 9\)
   (c) \(2.39 : 1\)

   Use von Neumann conditions at the boundary, if needed, and round the height to the nearest integer. Print out the resulting images and turn them in.

3. Resize, using cubic interpolation, the \(4 \times 4\) pixel image
   \[
   \begin{bmatrix}
   0.75 & 0.75 & 0.25 & 0.25 \\
   0.75 & 0.75 & 0.25 & 0.25 \\
   0.75 & 0.75 & 0.25 & 0.25 \\
   0.75 & 0.75 & 0.25 & 0.25 \\
   \end{bmatrix}
   \]
   into a \(100 \times 100\) pixel image. Use first order von Neumann boundary conditions at the boundaries of the image.
   (a) Print out the resulting image.
   (b) Write down the maximum and minimum intensities of the resulting image. Is the maximum larger than that of the original image? What about the minimum?

4. Use “bird.bmp” or “seal.bmp” or “lizard.bmp” in the following: Resize, using cubic interpolation, with first order von Neumann boundary conditions, to get a \(500 \times 500\) image. Print out the resulting image.