1. Considers the image below. What is the dimension of the matrix that represents the image? [10%]

2. If we filter using a 3x3 kernel and do not perform padding on the borders (convolution in the regions where filter and image are fully intersected), what will be the dimension of the previous image after filtering? [10%]

3. What will be the output dimension if the kernel is 5x5 (convolution in the regions where filter and image are fully intersected)? [10%]

4. Assuming the dimensions of the image are M x N. can you come out with a general formula that tells you the dimension of the image after filtering if only consider ‘valid’ regions (where filter and image fully intersect)? [10%]

5. Compute the output of applying the filter $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ on the first 5 valid columns of row 2 of “mario” (row 2 is the one that start with values 255 255 210 ..). Show your computations and write the obtained output. [10%]
Kernel estimation

\[ G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \]

6. Use the formula of 1D Gaussian function to find coefficients of a kernel of size 7 when \( \sigma = 1.4 \). Hint: \( x \) is evaluated in interval \([-3, -2, -1, 0, 1, 2, 3]\) [10%]

7. The size of a Gaussian kernel is usually chosen to have values in the order of 2 or 3 sigmas, since after that the values of the function are almost zero. In the extreme parts of this kernel (when \( x \) is either -3 or 3) how many sigmas it corresponds to? Is the chosen size of 7 a good value? [10%]

8. Approximate the obtained kernel as a fraction of integer numbers. Hint: use 64 as the denominator. [10%]

9. Compute a 7x7 Gaussian kernel using the 1D estimated kernel you estimated in the previous exercise. Remember, this is a separable filter and can be obtained using matrix multiplication. [10%]

\[ G = K_{7x1} \ast K_{1x7} \]

10. In class we build a sharpen filter as the sum of original filter + detail. The detail part was built with the original function and a box filter. Create a new kernel for sharpening but this time uses a Gaussian filter. [10%]