1. Write a program to perform a 1-D discrete Fourier transform of an array with at least \( N = 64 \) samples; for convenience, you can set \( N \) to be a power of 2, but this is not essential. For our purposes, it is most convenient to index the array over the domain \(-\frac{N}{2} \leq n \leq \frac{N}{2} - 1\), which is the interval \(-32 \leq n \leq +31\) if \( N = 64 \). In this case, the formula for the DFT is:

\[
F[k] = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} f[n] \exp\left(-2\pi i \frac{nk}{N}\right)
\]

where \( i \equiv \sqrt{-1} \). The output index \( k \) may be any integer, but is sensibly evaluated over the range:

\[-\frac{N}{2} \leq k \leq \frac{N}{2} - 1 \implies -32 \leq k \leq +31 \text{ if } N = 64\]

2. Use the program in #1 to evaluate the 1-D DFT of the following functions. NOTE: the output is complex, so you will need to plot arrays for the real part and for the imaginary part.

(a) \( f[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases} \) (result should be a constant)

(b) \( f[n] = \begin{cases} 1 & \text{if } |n| \leq 3 \\ 0 & \text{if } n > 3 \end{cases} \)

(c) \( f[n] = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n \neq 1 \end{cases} \) (result should be a cosine in the real part and a sine in the imaginary part)

(d) \( f[n] = \begin{cases} 1 & \text{if } n = 2 \\ 0 & \text{if } n \neq 2 \end{cases} \)

3. The 2-D DFT of the array \( f[n, m] \) is:

\[
F[k, \ell] = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f[n, m] \exp\left(-2\pi i \frac{(nk + \ell m)}{N}\right)
\]

where again the two sensible ranges of output indices is:

\[-\frac{N}{2} \leq k, \ell \leq \frac{N}{2} - 1 \text{ or } 0 \leq k, \ell \leq N - 1\]

This is usually evaluated by computing the 1-D DFT of each row individually (to get a generally complex valued result), followed by 1-D DFTs of each column individually, so you can use your 1-D DFT from #1 to implement the 2-D DFT.
4. Use the program in #2 to

(a) evaluate $F[k, \ell]$ for a rectangle function centered at the origin:

$$f[n, m] = \begin{cases} 1 & \text{if} \quad -3 \leq n, m \leq +3 \\ 0 & \text{if either} \quad |n| > 3 \quad \text{or} \quad |m| > 3 \end{cases}$$

In this case, the imaginary part should be zero, but may not be due to roundoff error. Plot the result as an image. Note that $F[k, \ell]$ will have negative values, so you will have to add a constant value to the result (or use TVSCL in IDL)

(b) OPTIONAL BONUS: apply a lowpass filter to $F[k, \ell]$ by removing any amplitude for values of $k, \ell$ that are larger than some value. For example, you could construct

$$G[k, \ell] = F[k, \ell] \cdot \begin{cases} 1 & \text{if} \quad -3 \leq k, \ell \leq +3 \\ 0 & \text{if either} \quad |k| > 3 \quad \text{or} \quad |\ell| > 3 \end{cases}$$

Then evaluate the inverse Fourier transform, which is identical to the program in #3 EXCEPT for a change in the sign of the exponent (and a trivial normalizing factor):

$$g[n, m] \equiv \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} f[k, \ell] \exp \left[ + \frac{2\pi i (nk + m\ell)}{N} \right]$$

Display the result as an image and submit.

(c) OPTIONAL BONUS: apply a lowpass filter to $F[k, \ell]$ by removing any amplitude for values of $k, \ell$ that are SMALLER than some value. Display the result and an image and submit.