MA3111: Mathematical Image Processing
Homework 3

To submit, zip the followings into mipXXXXX.zip where XXXXX is your student ID:

- **report file: contains answers to all questions, including all requested figures.**
- `sample.m` **and** `recover.m`**: code written for Problem 1.**
- `sharpen_by_frequency_laplacian.m`**: code written for Problem 2.**

Problem 1: Sample and recover 1-D continuous signal

In this problem, you will implement MATLAB functions `sample` and `recover` to sample from a 1-D continuous signal $x(t)$ at 1 Hz followed by recovering $x(t)$ from the samples $x_p(t)$. Then you can test your code using the provided `test.m`, where you will observe whether samples of sinusoids of different frequencies can be used to recover the original signal.

Question: suppose that $\exp(j2\pi\mu_0 t) \leftrightarrow \delta(\mu - \mu_0)$, what is the Fourier Transform of $x(t) = \sin(2\pi\mu_0 t)$?

For every second, we store and display 100 signal points.

Step 1:
Complete function `sample(x,N)`. It takes an input 1-D array `x` representing $x(t)$ and a parameter `N`, and only retains signal points at indices divisible by `N`. At all other indices the output signal should be zero. Since `N` will be set to 100 when `sample` is called, this effectively samples input signal $x(t)$ at 1 Hz sampling rate:

$$x_p(t) = \begin{cases} x(t) & \text{if } t \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}.$$

Step 2:
Complete function `recover(x_p,N)` using sinc function $sinc(x) \equiv \begin{cases} 1 & \text{if } x = 0 \\ \frac{\sin(\pi x)}{\pi x} & \text{otherwise} \end{cases}$ as following:

$$\hat{x}(t) = \sum_{k\in\mathbb{Z}} x_p(k) sinc(t-k).$$

Step 3:
Test your code by running `test.m`, where you set $x(t) = \sin(2\pi\mu_0 t)$. You should see a plot containing $x(t), x_p(t), \hat{x}(t)$. Try with `frequency=0.3` and `frequency=-0.7`, where `frequency` is μ_0. Include the resulting plots in the report. Is the Nyquist Sampling Theorem *roughly* correct, i.e., only the sinusoid with frequency 0.3 can be *roughly* recovered from its samples?

Bonus question: for `frequency=0.3`, does the recovered signal $\hat{x}(t)$ align perfectly with $x(t)$? Please explain why if it doesn't. Do they align better after increasing `duration_in_second` from 10 to 30? Explain.

Caution: Do not call MATLAB's built-in functions to sample and recover.

Problem 2: Frequency domain Laplacian sharpening

In this problem, you will implement frequency domain sharpening of gray level images using Laplacian. You will test your codes on the image `blurry-moon.tif` provided in HW2.

Fill in the computation in `sharpen_by_frequency_laplacian.m`, where you compute the output image

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y).$$

The difference is that $\nabla^2 f(x, y)$ is approximated by $\mathfrak{F}^{-1}(-4\pi^2(u^2 + v^2)F(u, v))$.

Hints: (feel free to ignore)

1. Use built-in functions `fft2` and `ifft2` for Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT), respectively.
2. Before doing anything, normalize $f(x, y)$ to the range $[0, 1]$.
3. Normalize $\nabla^2 f(x, y)$ to the range $[-1, 1]$ using `lf = lf / max(abs(lf(:)))`, where `lf` is $\nabla^2 f(x, y)$.

Try $c = 1$ and $c = -1$, and attach the corresponding output images in the report. However, if you do not follow the tips above to scale the image and its Laplacian, you may need to use different values of c to make the image look good.