Notes:
Fourier Transform is one of the most beautiful formulas in the world. If there is no Fourier Transform, there will be no computers, and no Pokemon Go.

Problem 1:
Consider the definition for Fourier Transform of \( f(x) \):

\[
\mathcal{F}[f](\lambda) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\lambda x} dx
\]

(a) Let

\[
f(x) = \begin{cases} 
1, & -a < x < a \\
0, & \text{else} 
\end{cases}
\]

where \( a \) is a positive number. Find the Fourier Transform of \( f(x) \).

(b) Let \( \mathcal{F}[f'](\lambda) = C \mathcal{F}[f](\lambda) \), \( C \) is a constant number. Use integration by parts to determine the constant number.

(c) Let’s consider the convolution. Let

\[
(f * g)(x) = \int_{-\infty}^{\infty} f(x-t) g(t) dt
\]

Let

\[
\mathcal{F}[f * g](\lambda) = C \mathcal{F}[f](\lambda) \mathcal{F}[g](\lambda)
\]

Determine the constant number. (Hint: Read the material after Theorem 6.6.1 in your book, the idea is the same, change the order of integral)

(d) In computers, we can use Fast Fourier Transform, which takes only \( O(N\log N) \) time for the discrete Fourier Transform of \( N \) data points. Since the inverse Fourier Transform in given by explicit formula (the time complexity is also \( O(N\log N) \)), we can compute convolution by Fast Fourier Transform to save computing time. Read the following website and try the example on it. You need to write down the output for clin and ccirc and compare with each other.