Discrete Fourier Transform (DFT) and Inverse DFT

Recall the Fourier Transform defined as:

Fourier Transform (FT): $$S(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}dt$$

Inverse Fourier Transform (IFT): $$s(t) = \int_{-\infty}^{\infty} S(f)e^{j2\pi ft}df$$

The FT allows a continuous-time signal, *s(t)*, defined over all time to be resolved into a continuous function of frequency, *S(f)*.
Conversely, the IFT allows a continuous-time signal, *s(t)*, to be recovered from its frequency domain representation, *S(f)*.
The so-called ***magnitude spectrum*** (often just called '***spectrum***') is defined as $|S(f)|^2$

In practice, *s(t)* is typically known only for some finite duration of interest, say $|t| \le T/2$. To apply Fourier analysis to such a finite duration signal, we must make the ***periodic extension assumption***. Call the periodically extended version $s_p(t)$. The complex exponential form of the Fourier series representation of $s_p(t)$ is:

$$s_p(t) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi nt/T} \qquad \text{where} \quad C_n = \frac{1}{T}\int_{-T/2}^{T/2} s(t)e^{-j2\pi nt/T}dt$$

Note - $s(t) = s_p(t)$ for $|t| \le T/2$

If only *N* discrete time samples of *s(t)* are known, i.e. $s(kT_s)$, $k = 0,1,2 \ldots N-1$ and T_s is the sampling period (Note: $T = N.T_s$) then the above Fourier series representation of the sequence of samples $s(kT_s)$ becomes:

$$s_p(kT_s) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi nkT_s/T} \qquad \text{where} \qquad C_n = \frac{1}{NT_s}\sum_{k=0}^{N-1} s(kT_s)e^{-j2\pi nkT_s/T}T_s$$

Normalising time to the sample period T_s, i.e. expressing time in samples instead of seconds, our Fourier series representation becomes:

$$s_p(k) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi nk/N} \qquad \text{where} \qquad C_n = \frac{1}{N}\sum_{k=0}^{N-1} s(k)e^{-j2\pi nk/N}$$

The function $e^{\pm j2\pi nk/N}$ is periodic in *k* with period *N* so we can ignore values of $s_p(k)$ outside the range $0 \le n \le N-1$.
The C_n term tells us the amplitude (and phase) of the *n*-th harmonic component which is exactly what we require from Fourier analysis. In practice we multiply across by *N* and use $N.C_n$ as the Discrete Fourier Transform (DFT). In practice *n* and *k* are interchanged as *n* is generally used as a time index and *k* as a frequency index and the DFT pair becomes:

DFT: $$S(k) = \sum_{n=0}^{N-1} s(n)e^{-j(2\pi/N)kn}$$

IDFT: $$s(n) = \frac{1}{N}\sum_{k=0}^{N-1} S(k)e^{j(2\pi/N)kn}$$

So given *N* discrete-time samples of a finite duration signal, we can evaluate *N* discrete-frequency samples of the spectrum of the original continuous-time signal.

Ex. 1. Write a Matlab or Octave function m-file to calculate and plot the DFT of an input sampled signal.
Ex. 2. Write a Matlab or Octave function m-file to calculate and plot the IDFT of an input sequence.