Homework 1 on the Fourier Transform

1. The divide-and-conquer step described in the lecture notes splits the terms of the sum for the discrete Fourier transform into odd and even terms. Construct a similar equation for use when $N = 3^n$ that divides the sum into three parts such that k (the index in the original sum) divided by 3 has remainder 0, 1 or 2.

2. Let $z = a + bi$ and $w = u + iv$ be complex numbers. It takes four real-valued multiplications when using the foil method to find the product zw. Look up fast complex multiplication, describe it and explain how many real-valued multiplications the fast algorithm uses to find zw.

3. Compute the number of real-valued double-precision floating-point multiplications and additions needed to perform a fast Fourier transform of length $N = 3^n$ using the divide-and-conquer step developed in your answer to Question 1. Explain your reasoning and how you counted the total number of operations. How many evaluations of the exponential function are performed?

4. Install the FFTW library `https://github.com/JuliaMath/FFTW.jl` in Julia and verify that it produces the same answers as the `myfft` routine we wrote in class. Test both routines for vectors of length $N = 2^n$ for at least 5 different vectors.

5. Modify the program `fft2.jl` so that it compares the speed of FFTW to the `myfft` routine. How much faster is FFTW compared to `myfft`? Does the difference in speed become more or less noticeable as the vector length N increases? If known, please provide details about the computer used, for example what model CPU it has.

6. [Extra Credit] Modify the code we wrote in class to perform fast Fourier transforms of length $N = 2^p 3^q$ where p and q are both non-negative integers. Do this by selecting the appropriate divide and conquer formula when dividing by the corresponding prime. Comparing the output of your routine with FFTW. Does it make any difference with regards to speed or rounding error if the powers of 3 or 2 are divided out first?