This is project work due at the end of the term.

A. In qGates.py, implement the following function.

```
def fourierRecursive(n):
    '''Assumes n >= 1. Returns the n-qbit quantum Fourier transform gate T.
    Computes T recursively rather than from the definition.'''
```

My implementation uses helper functions `fourierS`, `fourierR`, `fourierQ`, and `fourierD`, but yours is not required to.

You are not required to implement everything in terms of small gates. For example, my `fourierD` constructs $D^{(k)}$ as a diagonal $2^k \times 2^k$ matrix.

If you haven't implemented the original `fourier` function yet, you may not implement that function by simply calling this function. I want you to end up with two unrelated ways to compute the gate T, that we can test against each other.

B. In qGates.py, add a `fourierRecursiveTest(n)` function. It can be as simple as comparing the results of `fourier(n)` and `fourierRecursive(n)`, passing if and only if they are equal (up to numerical imprecision).