Today's assignment is project work. You will edit your ongoing library of Python functions. As always, the project is due at the end of the term, but you should complete the project work as it is assigned.

A. Download qAlgorithms.py from our course web site. Implement Bennett's key exchange algorithm in the function `bennett`. You'll need to perform a measurement of a one-qbit state. We don't currently have a function to do that. To work around this issue, I recommend that you tensor the one-qbit state with $|0\rangle$, then measure the first qbit of the resulting two-qbit state. It's not pretty, but it should work. Run the tests to help you fix errors.

B. In qGates.py, implement the function `function` based on the following specification. (By the way, we will generalize this function to more qbits later.)

```
def function(n, m, f):
    '''Assumes n, m == 1. Given a Python function f : {0, 1}^n -> {0, 1}^m.
    That is, f takes as input an n-bit string and produces as output an m-bit
    string, as defined in qBitStrings.py. Returns the corresponding
    (n + m)-qbit gate F.'''
```

C. In qAlgorithms.py, implement Deutsch's algorithm in the function `deutsch`. Run the tests to help you fix errors in `function` and `deutsch`.

D. In qGates.py, you already have a function `application`. In its doc string, change "`Assumes n == 1 or 2`" to "`Assumes n >= 1`". Improve it to match this new specification. (It is possible that your one- or two-qbit implementation is already sufficient.) Re-run the tests to make sure that they still work. Feel free to add new tests of your own.

E. In qGates.py, you already have a function `tensor`. Alter its doc string as follows. Then improve the implementation to match the new doc string. Re-run the tests. I recommend that you add new tests of your own. For example, it should not be hard to check $I \otimes H \otimes H$, $X \otimes I \otimes X$, etc.

```
def tensor(a, b):
    '''Assumes that n, m >= 1. Assumes that a is an n-qbit state and b is an
    m-qbit state, or that a is an n-qbit gate and b is an m-qbit gate. Returns
    the tensor product of a and b, which is an (n + m)-qbit gate or state.'''
```