This is a Python programming assignment. You will edit your ongoing copy of `qc.py` and submit it for grading. The grader will `import qc` and then run their own testing code against it. Probably the grader will also inspect your code.

For the sake of time, I am going to give you a bunch of code, that ordinarily I would ask you to write. I will send you this code in e-mail. Process that e-mail before you attempt this assignment.

A. Write a function according to the following specification. Your solutions to Homework 22 should help immensely.

```
def control(gate):
    '''Given an n-qbit gate U, returns the (n + 1)-qbit gate cU, in which the
    first qbit controls whether U is applied to the last n qbits.'''
```

B. Define constants `e7`, `ket07`, `ket17` that correspond to the encoding matrix and the encoded states $|\tilde{0}\rangle$, $|\tilde{1}\rangle$. By "encoding matrix" I mean the matrix E such that $|\tilde{0}\rangle = E\,|0000000\rangle$ and $|\tilde{1}\rangle = E\,|1111111\rangle$. Remember that E is not a gate (why?). Remember that the first argument to `application` must be a gate.

C. Write a function according to the following specification. Keep in mind that, even if a state is uncorrupted, floating-point imprecision might cause it to differ slightly from its mathematical ideal. Your implementation should be robust against small floating-point errors.

```
def decoding7(state7):
    '''Assumes that the given 7-qbit state is an uncorrupted encoding of a
    classical one-qbit state. Returns the one-qbit state.'''
```

D. Write a function according to the following specification.

```
def detection7(state7):
    '''Given a seven-qbit state that is the encoding of a one-qbit state, but
    possibly corrupted by one of the 22 errors. Implements the detection
    circuit. Returns a list or tuple consisting of seven elements: six
    classical one-qbit states (ket0 or ket1) and one seven-qbit state.'''
```

After completing this work, you should of course test your code. I suggest a function named something like `steaneTest7` that randomly picks either $|0\rangle$ or $|1\rangle$, encodes it, introduces an error using `error7`, and then detects and corrects that error.