Logbook 4 (Weeks 17-22)

https://github.com/ZMRamsey/APDCoursework.git
The full package, including a copy of this logbook, is stored in this git repository.

Modelling Circuits

```
>> OR*kron(NOT,AND)

ans =

     0     0     0     0     1     1     1     0
     1     1     1     1     0     0     0     1

fx >>
```

Quantum Computing

C		B		A
[0] [1]	<- H <-	$\frac{1}{\sqrt{2}}$ [1] = [0.707] [-1] [0.707]	<- H <-	[0] [1]
[1] [0]		$\frac{1}{\sqrt{2}}$ [1] [1]		[1] [0]

Probability model makes reversibility impossible

Correctness

```
public static int cube(int n) {
    int cube = 0, threeNsquared = 0, threeN = 0;
    int i = 0;

[1] assert cube == i*i*i : cube;
    assert threeN == 3*i : threeN;
    assert threeNsquared == 3*i*i : threeNsquared;

    while (i < n) {

[2]     assert cube == i*i*i : cube;
        assert threeN == 3*i : threeN;
        assert threeNsquared == 3*i*i : threeNsquared;

        cube = cube + threeNsquared + threeN + 1;
```

```
[3]     assert cube == (i+1)*(i+1)*(i+1);
        assert threeN == 3*i;
        assert threeNsquared == 3*i*i;

        threeNsquared = threeNsquared + (2*threeN) + 3;

[4]     assert cube == (i+1)*(i+1)*(i+1);
        assert threeN == 3*i;
        assert threeNsquared == 3*(i+1)*(i+1);

        threeN = threeN + 3;

[5]     assert cube == (i+1)*(i+1)*(i+1);
        assert threeN == 3*(i+1);
        assert threeNsquared == 3*(i+1)*(i+1);

        i++;

[6]     assert(cube == i*i*i);
        assert(threeN == 3*i);
        assert(threeNsquared == 3*i*i);
    }
[7] assert(cube == n*n*n);
    assert(threeN == 3*n);
    assert(threeNsquared == 3*n*n);
    assert(i==n);

    return cube;
}
```

[1] Cube is i^3, threeN is 3i, and threeNsquared is $3(i^2)$. These are the main assertions through the program to prove it is correct.
[2] See [1].
[3] Cube is now $(i+1)^3$ which is preparing for the increment of i.
[4] threeNsquared is now $3*(i+1)^2$ to prepare for the increment of i also.
[5] threeN is now 3*(i+1) again to prepare for the increment.
[6] i has been incremented, and so the assertions return to the main assertions from [1]
[7] The main assertions remain, but i is replaced with n to prove that the function returns the cube of n.

Self Assessment

Week	Score	Reasoning
Modelling Circuits	2	*A matrix has been created of the final result*
Quantum Computing	3	*There are matrices for the values at A, B, and C of the Hadamard gates*
Correctness	3	*The algorithm has been proven, with explanation of the assertions in place*
Complexity	0	*Not completed*