Coding exercise 2

a)

- Create a stack
- Iterate to take the reminder push on to the stack and divide the decimal by 2
- Another loop to pop everything from the stack

The popped version will store the correct sequence of the bits. (meaning the top of the stack is most significant and bottom is the least significant).

b)

The recursive solution doesn't seem that much different than the iterative solution. The problem being that the solution I came up with, for the recursive, ended up giving the answer in reverse order so I ended up making a reverse function to get the correct ordering of the bits. This problem was avoided in the iterative solution due to the nature of how stacks work. The time and space complexity is nearly the same. Though the iterative solution had the problem of needing to pop the stack to get access to all the bits. So they both have their fair share of pros and cons. I also noticed that it was easier to format the iterative solution than the recursive one. In that sense, I felt like the iterative solution was a bit more versatile and flexible compared to the recursive one. Not saying formatting wasn't possible for the recursive one; only that it would've taken a few extra steps.

c)

References

Adding to a existing string -
http://www.cplusplus.com/reference/string/string/append/

Rand num generator -
http://www.cplusplus.com/reference/cstdlib/rand/

Output Formatting -
http://www.cplusplus.com/reference/iomanip/setw/
http://www.cplusplus.com/reference/ios/left/

Coding exercise 3

I did notice a few differences from the self-made and the developer made stack library. For example, the differences in constructors. I noticed that I couldn’t declare a fixed size for the stack using the constructor. This threw me off a bit due to the fact that I wanted to make the stack be 32 since there are 32-bits in an unsigned integer (why unsigned integer? Developer's decision aka me). Another notice I took was that the pop function was void. It didn’t return anything unlike the one that was made in the self-made library. If I wanted to pop but also retrieve what I popped I’d need to use the top function for the developer made stack library. The only function that felt consistent for both libraries was the push function.