1. Problem 2.2

2. Problem 2.7 The instructions are a bit confusing - what it means is that you should determine how long a “song script” would be if you had some clever way of expressing the repeats. You can think about this as the “encoding” of the full song. The length of this “encoding” is your $f(n)$.

For example, suppose the original lyrics has a total number of $k$ lines. Some (not very clever) “encoding” looks like this:

```plaintext
Function 12daysofXmas()
    lines[1] = original lyrics line 1
    ...  
    lines[k] = original lyrics line k  
    for $i = 1$ to $k$ do
        output lines[$i$]  
    end
```

Recall that the lines in the original lyrics are bounded by some constant $c$, thus the length of the original lyrics is $kc$. This encoding clearly has the same number of lines as the original, plus the 3 additional lines in the `for` loop. The initialization lines each has length $c_1 + c$, where $c_1$ is another constant that denotes the length of “`lines[*] = `”. Note $c_1 < c$. The length of the lines in the `for` loop is constant, and shorter than the original lyric lines and is thus $< c$. Therefore the length of the encoding is $f(n) < (k(c_1 + c) + 3c) = kc + kc_1 + 3c$, where $n = kc$. Thus $f(n)$ is $O(n)$.

Please include description of your encoding scheme and pseudo code - it should be clear that your encoding must be length-reducing, i.e. $O(n)$ is not acceptable. Focus on the length analysis. You may skip proof of correctness.

3. Problem 2.8a Full write-up. 2.8b is tricky and is extra credit. If you don’t have all of it, try to sketch out what you understood to be the key points and which direction might be right.

4. Problem 3.2 Full write-up.

Please hand in your assignment electronically on Moodle.