Problem 3: Implementing Many-Time Pad

The many-time pad implementation is a direct derivative of the one-time pad. Despite being nearly the same, it creates a major weakness. The only meaningful difference needed was to generate a key suitably long enough for all messages. The plaintext messages are submitted and then a key long enough to encrypt the longest plaintext message is generated. The encryption step uses the XOR operation between the key and the message, in turn creating the ciphertext. The same key is used for all messages; if a message is shorter than the key, then the key and the plaintext apply XOR operation for the length of the plaintext message.

The most obvious result is that plaintext characters (and words) in the same position between messages are encrypted into the same ciphertext. Spaces are especially common and a strong weakness.

The messages:

```
"and i ran out of text already"
"and so on"
"and on"
```

Result in ciphertext (in hex):

```
529ffc8892077def01d88b4fc10f4f4c416b8081a7da4207848edaae7c
529ffc8888482fe101
529ffc889449
```

This is leads to the understanding that the XOR of the key at any given position yields the same result. If we XOR those two ciphertexts together, it reveals areas of matching words and replicates the same pattern as the same two plaintexts would produce when an XOR is performed on them. When 10 ciphertexts are all analyzed the same way, many insights into the similarities (and differences) between the messages becomes apparent, and so guessing the contents becomes possible; thus the crib drag is born. Additionally, frequency analysis can help give guesses direction.