**D E A R  J E N**

This year your birthday presents will be obtained through a small treasure hunt, comprised of four sequential parts outlined by the clues below. Your final (and main) present is in the lock box you've just received; to find the key you will need to complete the whole treasure hunt.

**Part 1**

To find the first present, you must think like Russell and indeed act like Russell. Particularly Russell first thing in the morning after he gets out of bed.

**Part 2**

To determine the location of the second present, complete your face and turn it over (this will make more sense after finding the first present).

**Part 3**

Your third present is in the spare room cupboard, but you will find entry to this cupboard quite impossible thanks to a masterful piece of engineering by your brilliant boyfriend. To bypass it, you will need to multiply together the following numbers:

- The last house number on ?
- The number of benches in ??
- The width of ??? to the nearest foot

**Part 4**

You're going to hate me for this one, but here we are anyway. The location of the key for the lock box is given by the clue "mrah gahm: fujonp rju mkunum". This is an encrypted version of a normal english sentence. The means of encryption is a simple one (and a fairly shit one, if truth be told), of my own design:

- Each character is replaced by its index in the alphabet (a: 0, z: 25 etc)
- The index is multiplied by an encryption key
- The resultant value is divided by 26 (because alphabet)
- The remainder of this division operation is mapped back to a character using the same indexing as in the first step - this is the encrypted character.

Pseudo-mathematically: $i(encrypted) = (i(unencrypted) \times key) \bmod 26$

By now you have the encryption key, so all you need to do is write an algorithm to reverse this encryption and run it on the sentence above. If you want to bypass some of the mundane setup faff required for this, I have created a code repo at https://github.com/rcarswell/happy_birthday_schmerfel which is missing only the implementation of the decryption method.

**H A P P Y  B I R T H D A Y**