**Question 1:**

**In the Binary Search algorithm, it is suggested to calculate the mid as beg + (end - beg) / 2 instead of (beg + end) / 2. Why is it so?**

**Answer:**

Because (left + right) may overflow. Which then means get a result that is less than left. Or far into the negative if using signed integers.

So instead, they take the distance between left and right and add half of that to left. This is only a single extra operation to make the algorithm more robust.

Suppose 'low' and 'high' are 16-bit unsigned integers. That means, they can only have a maximum value of 2^16=65536. Consider this, low = 65530 high = 65531

If added them first, (low + high) would end up being junk since that big a number (131061) cannot be stored in a 16-bit integer. And so, mid would be a wrong value.