**Extra Credit 1.**

• • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •

**Question 1.** Ask the user to enter a random binary array having the first *K* numbers equal to *0* and the rest equal to *1*. Write a function to find the position of *K* that splits the 0s and 1s. The time complexity of your solution should be ***O(logn)*.**)

(*Example*: input: a = [ 0 0 0 1 1] ➔ output: K = 3)

**Question 2.** Given *two sorted* arrays each of size *n*. Find the median of an array resulting from merging the two arrays. (Hint: You could use the same approach of binary search algorithm. The time complexity of your solution should be ***O(logn)*.**)

*Example 1*:
a1 = [0, 2, 10, 26, 68], >> median = 10
a2 = [1, 11, 18, 20, 41], >> median = 18
*Output*: Median = (11+18)/2 = 14.5

*Example 2*:
a1 = [5, 6, 14, 26], >> median = (6+14)/2 = 10
a2 = [3, 41, 88, 100] >> median = (41+88)/2 = 64.5
*Output*: Median = (14+26)/2 = 20

*Example 3*:
a1 = [5, 10],
a2 = [2, 41]
*Output*: Median = **{max(a1[0], a2[0]) + min(a1[1], a2[1])}/2** = {max(5,2)+min(10,41)}/2 = {5+10}/2 = 7.5