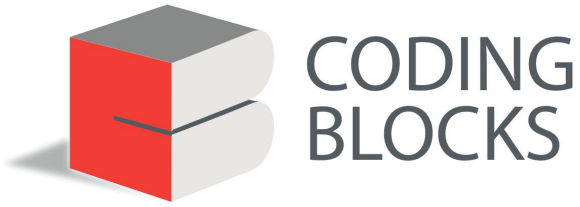

Recursion-2

Batch: Crux

1. Implement Binary Search Recursively
2. Given a String Return all the subsequences. e.g. for input = abc you need to print “”, a, b, c, ab, ac, bc, abc
    a. Instead of returning print all these (make a print function, it will not return anything and will just print all subsequences)
3. Using the phone keypad return all possible words that can be produced given input digits. e.g. 23 > “ad, ae, af, bd, be, bf, cd, ce, cf”
    a. Instead of returning print all these (make a print function i.e it will not return anything and will just print all combinations)
4. Suppose you have a string made up of only the letters 'a' and 'b'. Write a recursive function that checks if the string was generated using the following rules:
    a. the string begins with an 'a'
    b. each 'a' is followed by nothing or an 'a' or "bb"
    c. each "bb" is followed by nothing or an 'a'
5. Return all permutations of a String
    a. Instead of returning print these (make a print function i.e it will not return anything and will just print all permutations)
6. Return all subsets of an array
    a. Instead of returning print all these

7. Given an array find all subsets of A which sum to K.
8. Assume that value of a=1, b=2, c=3, d=4, ... z=26. You are given a numeric string S. Write a program to find and print list of all possible codes that can be generated from the given string. E.g. 1123 aabc, kbc, alc, aaw, kw
9. A child is running up a staircase with n steps, and can hop either 1 step, 2 steps or 3 steps at a time. Implement a method to count how many possible ways the child can run up to the stairs.