1. Add Binary
   Given two binary strings a and b, return their sum as a binary string.
   Example 1: Input: a = "11", b = "1" Output: "100"
   Example 2: Input: a = "1010", b = "1011" Output: "10101"
2. Find the Index of the First Occurrence in a String
   Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.
   Example 1: Input: haystack = "sadbutsad", needle = "sad" Output: 0
   Explanation: "sad" occurs at index 0 and 6.
   The first occurrence is at index 0, so we return 0.

   Example 2: Input: haystack = "leetcode", needle = "leeto" Output: -1
   Explanation: "leeto" did not occur in "leetcode", so we return -1.
3. Reverse Vowels of a String
   Given a string s, reverse only all the vowels in the string and return it.

   The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in both lower and upper cases, more than once.
   Example 1: Input: s = "hello" Output: "holle"
   Example 2: Input: s = "leetcode" Output: "leotcede"
4. Decode String
   Given an encoded string, return its decoded string.

   The encoding rule is: k[encoded_string], where the encoded_string inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer.

   You may assume that the input string is always valid; there are no extra white spaces, square brackets are well-formed, etc. Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k. For example, there will not be input like 3a or 2[4].

   The test cases are generated so that the length of the output will never exceed 105.
   Example 1: Input: s = "3[a]2[bc]" Output: "aaabcbc"

   Example 2: Input: s = "3[a2[c]]" Output: "accaccacc"

   Example 3: Input: s = "2[abc]3[cd]ef"
   Output: "abcabccdcdcdef"