1. Compare Version Numbers.
   Given two version numbers, version1 and version2, compare them. Version numbers consist of one or more revisions joined by a dot '.'. Each revision consists of digits and may contain leading zeros. Every revision contains at least one character. Revisions are 0-indexed from left to right, with the leftmost revision being revision 0, the next revision being revision 1, and so on. For example 2.5.33 and 0.1 are valid version numbers.

   To compare version numbers, compare their revisions in left-to-right order. Revisions are compared using their integer value ignoring any leading zeros. This means that revisions 1 and 001 are considered equal. If a version number does not specify a revision at an index, then treat the revision as 0. For example, version 1.0 is less than version 1.1 because their revision 0s are the same, but their revision 1s are 0 and 1 respectively, and 0 < 1.

   Return the following:

   If version1 < version2, return -1.
   If version1 > version2, return 1.
   Otherwise, return 0.
   Example 1: Input: version1 = "1.01", version2 = "1.001" Output: 0
   Explanation: Ignoring leading zeroes, both "01" and "001" represent the same integer "1".

   Example 2: Input: version1 = "1.0", version2 = "1.0.0" Output: 0
   Explanation: version1 does not specify revision 2, which means it is treated as "0".
   Example 3: Input: version1 = "0.1", version2 = "1.1" Output: -1
   Explanation: version1's revision 0 is "0", while version2's revision 0 is "1". 0 < 1, so version1 < version2.

2. Count Special Integers.

   We call a positive integer special if all of its digits are distinct.

Given a positive integer n, return the number of special integers that belong to the interval [1, n].

Example 1: Input: n = 20 Output: 19

Explanation: All the integers from 1 to 20, except 11, are special. Thus, there are 19 special integers.

Example 2: Input: n = 5 Output: 5

Explanation: All the integers from 1 to 5 are special.

Example 3: Input: n = 135 Output: 110

Explanation: There are 110 integers from 1 to 135 that are special.

Some of the integers that are not special are: 22, 114, and 131.

3. longest ideal sequence.

   You are given a string s consisting of lowercase letters and an integer k. We call a string t ideal if the following conditions are satisfied: t is a subsequence of the string s.

The absolute difference in the alphabet order of every two adjacent letters in t is less than or equal to k. Return the length of the longest ideal string. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

Note that the alphabet order is not cyclic. For example, the absolute difference in the alphabet order of 'a' and 'z' is 25, not 1.

Example 1: Input: s = "acfgbd", k = 2 Output: 4

Explanation: The longest ideal string is "acbd". The length of this string is 4, so 4 is returned. Note that "acfgbd" is not ideal because 'c' and 'f' have a difference of 3 in alphabet order.
Example 2: Input: s = "abcd", k = 3 Output: 4
Explanation: The longest ideal string is "abcd". The length of this string is 4, so 4 is returned.

4. . Roman to Integer

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9.
X can be placed before L (50) and C (100) to make 40 and 90.
C can be placed before D (500) and M (1000) to make 400 and 900.
Given a roman numeral, convert it to an integer.

Example 1: Input: s = "III" Output: 3
Explanation: III = 3.

Example 2: Input: s = "LVIII"
Output: 58
Explanation: L = 50, V= 5, III = 3.
Example 3: Input: s = "MCMXCIV"
Output: 1994
Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.