STRINGS

1. *int count(char* s, char c)* that counts how many times the character *c* occurs inside string *s*.
2. Count the number of digits in a string. For example, "abcd12hjk3425kk" has 6 digits.
3. *Char*repeat(char*s, int n)* that creates and returns a new string that is made by concatenating n copies of the parameter string s. For example, calling this method with the parameters "Hello" and 3 would return the string "HelloHelloHello", if n equals 0 function should return empty string.
4. A simple string contains a large repetition of letters within it. This problem is related to string handling and manipulation. An original message is sent from planet Earth to Cybertron in form of a string. However, the letter position and string size is not important. The number of time each letter has occurred in the string is important. So the original string which is sent to Cybertron is encrypted in the new string which comprises the letters followed by each time it has occurred in the original string.
   Ex: original message is abcdabf. Then the encrypted string is a2b2c1d1f1.**Input**: The input consists of a single line string without any space or numeric or special characters. **Output**: It will consist of in the encrypted string which comprises the letters followed by each time it has occurred in the original string in order. Ex: **Input:** information**, Output**: i2n2f1o2r1m1a1t1.
5. Given three strings A, B and C. Write a function that checks whether C is an interleaving of A and B. C is said to be interleaving A and B, if it contains all characters of A and B and order of all characters in individual strings is preserved.
6. Write a program that inputs a sequence and counts the number of palindromes in the sentence.
7. Given a string and a positive integer d. Some characters may be repeated in the given string. Rearrange characters of the given a string such that the same character become d distance away from each other. Note that there can be many possible rearrangements; the output should be one of the possible rearrangements. If no rearrangement possible report that also as an error. Write an efficient code for this.
   For example: INPUT: "abb" d = 2. Output = "bab".
8. Implement strStr().
   Strstr- locate a substring (needle) in a string (haystack). **[FACEBOOK][AMAZON][MICROSOFT]**
   Returns the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.
   Note: Good clarification questions:
   What should be thre return value if the needle is empty?
   What if both haystack and needle are empty?
   For the purpose of this problem, assume that the return value should be -1 in both cases.
9. Given a string S consists of upper/lower-case alphabets and empty space characters ' ', return the length of last word in the string. If the last word does not exist, return 0. [**AMAZON]**
   Note: A word is defined as a character sequence consists of non-space characters only.
   Example: Given S = "Hello World",
   Return 5 as length ("world") = 5.
   Please make sure you try to solve this problem without using library functions. Make sure you only traverse the string once.
10. Given a roman numeral, convert it to an integer. Input is guaranteed to be within the range from 1 to 3999.
    Example: **[TWITTER] [Intuit]**
    INPUT: "XIV" return: 14
    "XX" return: 20
11. Write a function to find the longest common prefix string amongst an array of string. Longest common prefix for a pair of strings S1 and S2 is the longest string S which is the prefix of both S1 and S2. **[GOOGLE]**
    Given the array as ["abcdefgh", "aefghijk", "abcefgh"] the answer should be "a".
12. Given a string containing only digits, restore it by returning all possible valid IP address combinations. A valid IP address must be in the form of A.B.C.D, where A,B,C and D are numbers from 0-255. The numbers cannot be prefixed unless they are 0.
    Example: "25525511135", return **[AMAZON]**
    ["255.255.11.135" ,"255.255.111.35"]. (Make sure the returned strings are sorted in order).

13. Implement AtoI to convert a string to an integer. **[NVIDIA][APPLE]**
    Example: Input: “9 7068”
    Output: 9
    There can be multiple test cases and doubts first try to think by yourself and then after that discuss with your mentor.
    Note: No use of library functions.
14. You are given a string. The only operation allowed is to insert characters in the beginning of the string. How many minimum characters are needed to be inserted to make the string palindrome string?
    Example: INPUT: ABC
    OUTPUT: 2
    INPUT: AACECAAAA
    OUTPUT: 2
15. Given two binary strings, return their sum also a binary string. **[FACEBOOK]**
    Example: a = “100”
    B = “011”
    Output: “111”
16. Find if given number is power of 2 or not. More specifically, find if given number can be expressed as 2^k where k>=1.
    Input: number length can be more than 64, which mean number can be greater than 2^64 (out of long range). **[AMAZON]**
    Output: return 1 if the number if a power of 2 else return 0.
17. Given two numbers represented as a string, return multiplication of the numbers as a string.
    Note: the numbers can be arbitrarily large and non-negative. **[MICROSOFT][GOOGLE]**
    Note: Your answer should not have leading zeroes. For example, 00 is not a valid answer.
    For example, given strings “12”, “10”, your answer should be “120”.
18. The count and say sequence is the sequence of integers beginning as follows:
    1, 11, 21, 1211, 111221,.. **[FACEBOOK]**
    1 is read off as one 1 or 11.
    11 is read off as two 1’s or 21.
    21 is read off as one 2, then one 1 or 1211.
    Given an integer n, generate the nth sequence.
    Note: the sequence of integers will be represented as a string.
    Example: if n = 2, the sequence is 11
19. Given an input string, reverse the string word by word. **[AMAZON][CISCO]**
    Example: given s = “the sky is blue”.
    Return “blue is sky the”.
    “
    A sequence of non-space characters constitutes a word.
    Your reversed string should not contain leading or trailing spaces, even if it is present in the input string.
    If there are multiple spaces between words, reduce them to a single space in the reversed string.
    ”
20. Given an array of words and a length L, format the text such that each line has exactly L characters and is fully (left and right justified). You should pack your words in a greedy approach; that is pack as many words as you can in each line. Pad extra spaces when necessary so that each line has exactly L characters. Extra spaces between words should be distributed evenly as possible. If the number of spaces on line does not divide evenly between words, the empty slots on the left will be assigned more spaces than the slots on the right. For the last line of text, it should be left justified and no extra space sis inserted between words. Your program should return a list of strings, where each string represents a single line. **[LINKEDIN][GOOGLE]**
    Example: words [“This”, “is”, “an”, “example”, “of”, “text”, “justification.”]
    L: 16 Return the formatted lines as:
    [“This is an”
    “example of text”,
    “justification. ”]
    “**Note:** each word is guaranteed not to exceed L in length.”