**Comcast Technical Interview Questions**

**1. Find all palindromic decompositions of a given string s.**

A palindromic decomposition of string is a decomposition of the string into substrings, such that all those substrings are valid palindromes.

**Example**

Input: "abracadabra"

Output: [ "a|b|r|a|c|a|d|a|b|r|a", "a|b|r|a|c|ada|b|r|a", "a|b|r|aca|d|a|b|r|a" ]

**Notes**

Input Parameters: There is only one argument: string s.

Output: Return array of string res, containing ALL possible palindromic decompositions of given string. To separate substrings in the decomposed string, use '|' as a separator between them.

• You need not to worry about the order of strings in your output array. Like for s = "aa", arrays ["a|a", "aa"] and ["aa", "a|a"] both will be accepted.

• In any string in your returned array res, order of characters should remain the same as in the given string. (i.e. for s = "ab" you should return ["a|b"] and not ["b|a"].)

• Any string in the returned array should not contain any spaces. e.g. s = "ab" then ["a|b"] is expected, ["a |b"] or ["a| b"] or ["a | b"] will give the wrong answer.

**Constraints:**

• 1

• s only contains lowercase letters ('a' - 'z').

Any string is its own substring.

**2. Given a variety of coin types defining a currency system, find the minimum number of coins required to express a given amount of money. Assume an infinite supply of coins of every type.**

##### Example

Input: Coin types: [1, 3, 5]. Amount to express: 9.

Output: 3

Here are all the unique ways to express 9 as a sum of coins 1, 3 and 5:

1, 1, 1, 1, 1, 1, 1, 1, 1

1, 1, 1, 1, 1, 1, 3

1, 1, 1, 1, 5

1, 1, 1, 3, 3

1, 3, 5

3, 3, 3

Last two ways use the minimal number of coins, 3.

##### Notes:

Every input will include a coin of value 1. This guarantees that a solution will always exist.

There will be no duplicate coin types in the input.

##### Constraints:

● 1

● 1

● 1

**3. Sort a given singly linked list in ascending order.**

**Input Format:**

There is only one argument named head, denoting the head of the given singly linked list.

**Output Format:**

After sorting, return the head of the same linked list that is provided in the input.

**Constraints:**

* 0 <= length of the list <= 10^5
* Nodes will contain integer values.
* You have to do it in-place (you must not create any new node)

‍**Sample Test Case 1:**

Sample Input 1:

1 -> 7 -> 4 -> 2 -> NULL

Sample Output 1:

1 -> 2 -> 4 -> 7 -> NULL

**Sample Test Case 2:**

Sample Input 2:

3 -> 2 -> 1 -> 5 -> 4 -> NULL

Sample Output 2:

1 -> 2 -> 3 -> 4 -> 5 -> NULL

**4. Write a code to convert a given set of integers into their Roman number equivalents.**

**Input:** num = 3

**Output:** "III"

**Explanation:** 3 is represented as 3 ones.

**5. Write a program to find out if a given number “N” is sparse. (A number is said to be sparse if no two bits are in binary representation).**

Input: x = 72

Output: true

Explanation: Binary representation of 72 is 0**1**00**1**000.

There are no two consecutive 1's in binary representation

Input: x = 12

Output: false

Explanation: Binary representation of 12 is **11**00.

Third and fourth bits (from end) are set.

**6. Write a program to find the lowest common ancestor of two nodes of a given binary tree “B” with unique values.**

**Input:**

n1 = 2 , n2 = 3

  1

  / \

  2 3

**Output:** 1

**Explanation:**

LCA of 2 and 3 is 1.

**7. Given a binary tree “B” with unique values, write a program to find:  1. The longest consecutive sequence. 2. The length of the longest path comprising connected nodes with consecutive values.**

10

/ \

/ \

11 9

/ \ /\

/ \ / \

13 12 13 8

Maximum Consecutive Path Length is 3 (10, 11, 12)

**Note**: 10, 9 ,8 is NOT considered since

the nodes should be in increasing order.

**8. Given a sequence, return its next lexicographically greater permutation. If such a permutation does not exist, then return it in ascending order.**

**For example**, lexicographically next permutation of “gfg” is “ggf” and the next permutation of “acb” is “bac”.   
**Note:** In some cases, the next lexicographically greater word might not exist, e.g, “aaa” and “edcba”.

**9. You are given alphanumeric strings s and t. Find the minimum window (substring) in s, which contains all the characters of t.**

**Input:** string = “this is a test string”, pattern = “tist”   
**Output:** Minimum window is “t stri”   
**Explanation:** “t stri” contains all the characters of pattern.

**Input:** string = “geeksforgeeks”, pattern = “ork”   
**Output:** Minimum window is “ksfor”

**10. You are given an array of integers, arr, of size n, which is analogous to a continuous stream of integers input. Your task is to find K largest elements from a given stream of numbers**

Input:

stream[] = {10, 20, 11, 70, 50, 40, 100, 5, ...}

k = 3

Output: {\_, \_, 10, 11, 20, 40, 50, 50, ...}